AN ALGORITHMIC/HEURISTIC METHODOLOGY FOR THE OPTIMUM UTILIZATION OF TECHNOLOGICAL INNOVATIONS AS APPLIED TO DIAMOND COATING

By

RICHARD KEITH STANFILL

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1995

ACKNOWLEDGEMENTS

It is my privilege to acknowledge the help and guidance received from many individuals during my doctoral studies. I am indebted to my chairman, Professor Ali Seireg, for his patience, encouragement and consistently invaluable advice. His vision and mastery of mechanical systems are a constant source of inspiration. I would like to thank Dr. Jack Elzinga for developing my understanding of the manufacturing system, serving on my doctoral committee and being a good coach. Dr. Carl Crane deserves great thanks for being an excellent supervisor for my graduate assistantships in the SUCCEED program. Dr. Crane's hands-off management style gave me great latitude while developing and delivering the Design and Build curriculum—a luxury few research/teaching assistants are allowed. I owe a thank you to Dr. Doug Dankel and Dr. Scott Smith for serving on my committee and being excellent teaching role models. I would also like to acknowledge the help from Dr. Rajiv Singh for sharing his resources and providing me with details of the diamond coating process.

Thanks also go to Mr. Jose Rodriguez for taking the leap from industry with me. Jose's vision and collaboration have led to many successful outcomes, chief among them our Design and Build course. I would like to thank Ms. Angela Maria Ventura-Medyk for all her help during the Design and Build program. Without her perseverance, we never would have gotten the program running. I would also like to acknowledge the help from the office staff in the Center for Intelligent Machines and Robotics, the Department of Mechanical Engineering and the Department of Electrical Engineering.

And lastly, I would like to thank my wife Linda for all her loving support, encouragement, patience and proof-reading prowess during my doctoral studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree Doctor of Philosophy

AN ALGORITHMIC/HEURISTIC METHODOLOGY FOR THE OPTIMUM UTILIZATION OF TECHNOLOGICAL INNOVATIONS AS APPLIED TO DIAMOND COATING

by

RICHARD KEITH STANFILL

August 1995

Chairman: Dr. Ali A. Seireg
Major Department: Mechanical Engineering

This research proposes a systematic procedure for the effective transfer of laboratory discoveries to the parametric design of new mechanical products or the improvement of existing products. The methodology is based on an associative object-oriented knowledge base that incorporates information on the new technology's features, attributes, cost and potential use. An expert system matches the new technology with potential applications. The resulting application matches are then optimized for life-cycle design criteria. Lastly, cost-benefit analyses of the different product applications are compared and ranked. The highest ranked application indicates where the greatest return on investment is possible for further development of the new technology.

A recent University of Florida laboratory discovery for diamond coating of tribological surfaces serves as an illustration of the methodology. This process shows promise as a practical surface treatment for tribological applications. The methodology is demonstrated for the case of gear pairs with different materials, center distance and gear ratio.

CHAPTER 1
INTRODUCTION

1.1 Research Motivation

Efficient transfer of laboratory discoveries into product applications is essential for current and future competitiveness in the world marketplace. This transfer process is very difficult because it requires intimate knowledge of the available technology as well as the needs of the marketplace. Market research is a well established industrial activity. The work in this area, however, is generally undertaken by business-oriented staff with a casual knowledge of what technological development can accomplish. In other words, marketing is generally based on *market pull* without consideration for *technology push*.

Critical components of this transfer process are the assessment of economic benefit of a new technology and selection of appropriate product applications. Industry needs a tool set to assess the financial risk associated with research and development of new technologies. A formal methodology for a domain-independent laboratory transfer process will help industry open new markets and develop new or improved products.

Noted Harvard economist John Kenneth Galbraith [Wein87d] sees the market power in the hands of the producer:

> The myth of economics is the sovereign consumer, but the reality of life is the producer who reaches out to take control, to influence the consumer, and to establish a mode of life serviceable to the producer. This mode is one in which the emphasis is on consumer achievement, which has always been at the heart of producer power (p. 52).

Galbraith further states "Beyond technology comes good design. . . so one part of the solution [to rejuvenating our older, more expensive work force] is to emphasize good design, along with good technology," (p. 51). He also points out that "the engineer brings in to

existence products that have their justification in being unique. And therefore engineers are to some degree monopolistic. . . The market can give a special power to the producer of a unique product," (p. 52).

1.1.1 Importance: Technology as a Worldwide Competitive Product

Efficient technology transfer has become a critical need in revitalizing industry's position in world markets. To survive, products must be competitive inside and outside of national borders. Seymour Baron [Baro90], associate director for applied programs at Brookhaven National Laboratory, points out:

> Indeed, competition is not viewed anymore between General Motors and Ford or between GE and Westinghouse but rather between our auto industry and the Japanese auto industry and our high technology versus those in Europe and the Far East. The market is not a U.S. market but a world market, and technology is a worldwide competitive product (p. 38).

From a more global perspective, the economic stability of entire nations may rest in successful transfer of technology from defense-related research into commercial products. The emerging independent states from within the former Soviet Union have a wealth of research talent previously devoted to military activities. The U.S. Department of Energy national laboratories and various U.S. industry partners have recognized this talent and have proposed joint programs to assist in redirecting these research institutions toward productive nonmilitary applications [Hnat93]. Congress has passed legislation—the FREEDOM Support Act—to provide U.S. aid to "facilitate the conversion of military technologies, capabilities and defense industries to civilian applications. . . and activities which will assist the independent states to transition to market economies and become responsible members of the global economic community," (p. 1).

1.1.2 Barriers to Technology Transfer

Seymour Baron [Baro90] outlines the hurdles to technology transfer from the national laboratories to industry. Prominent among these hurdles is that

> laboratories. . . do research for research's sake and not with any product in mind. Results are reported openly and discussed fully at conferences. . . no cost-benefit analysis to determine whether the research will be profitable. . . Industry. . . approaches research mainly as short term missions for reasonable return on investment (p. 39).

This insensitivity to economic issues directly affects the university-lab-to-industry technology transfer process. The deficiency can be overcome by developing a tool for laboratories to perform systematic, quantitative cost/benefit analyses. In order to entice industry to participate in the technology transfer process, laboratories need the ability to demonstrate better return on investment (ROI) for their process over traditional approaches.

1.2 Framework of Proposed Methodology

The proposed methodology is a two-stage process. The first stage is a heuristics-driven knowledge base management system for qualitatively matching new technologies to potential product applications. The knowledge base management system provides the second stage with discovery-application matches. Stage two is an algorithmic module for quantitative evaluation of the match and for producing cost/benefit analyses. After multiple discovery-application matches are evaluated, a ranking of the applications for the new technology can be produced. Ultimately, these modules will interact with one another inside a common, intelligent framework. A conceptual depiction of the system is shown in Figure 1-1.

1.2.1 Plausible Application Selection

Plausible product applications for emerging technologies are selected within three interacting modules of a knowledge-based system. A knowledge base containing pertinent physical and process information for new technologies interacts with a knowledge base containing product applications—including current limitations and desired capabilities. A

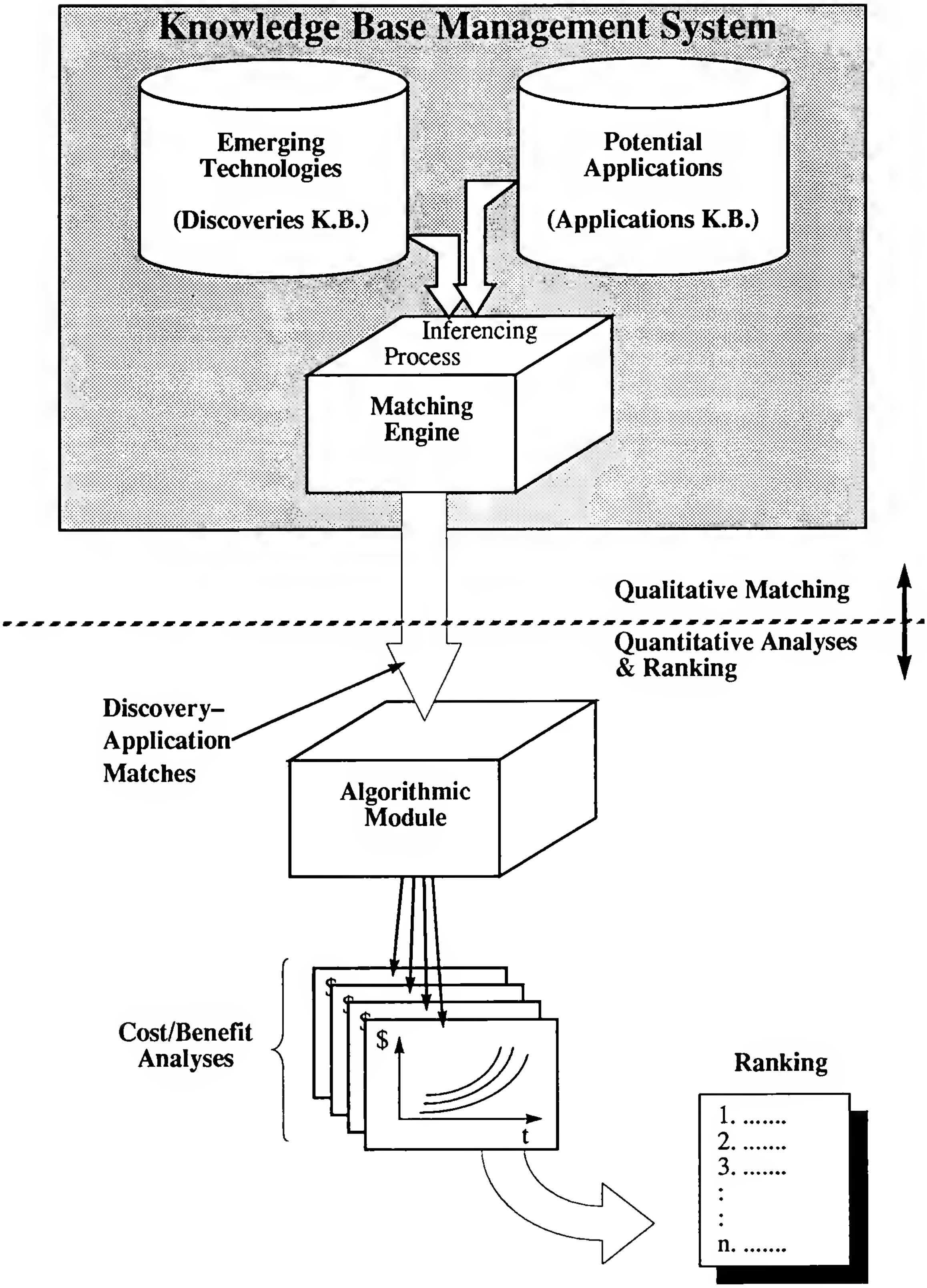


Figure 1-1. The technology transfer framework for the proposed methodology.

third module uses an inferencing strategy to match potential product applications with new technology processes. The product of these three interacting modules is a ranked list of plausible product applications.

1.2.1.1 New technology knowledge base

The knowledge base for the new technologies contains a structured representation of process information. Knowledge regarding the new technologies' features, attributes, costs, physics, processing equipment involved, etc. is encoded in the knowledge base.

1.2.1.2 Needs and applications knowledge base

Knowledge regarding existing product domains is captured in a needs and applications knowledge base. This information includes product features, attributes, costs, physics, current processing constraints, desired capabilities and processing equipment involved.

1.2.1.3 Inference engine/pattern matcher

The inference engine searches the new technologies knowledge base and the needs and applications knowledge base for potential application matching. The new technology and application matched pairs are provided in a ranked list. Heuristics in the inferencing module produce a qualitative ranking.

1.2.2 Algorithmic Analyses of Plausible Applications

Qualitative discovery-application matches are passed to an algorithmic analysis module for evaluation. The module gathers relevant functional relationships and design parameters for each match. An objective of maximum cost/benefit is formulated and the design space is bound by user-provided constraints. A simulation—utilizing the functional relationships gathered previously—is run to generate a series of feasible designs. The results are then post-processed for cost/benefit.

1.2.3 Cost/benefit Analysis

Before the discovery-application matches can be ranked, the results of the algorithmic module are post-processed. The results of this activity are a series of graphs relating the

cost of implementing the discovery versus increase in performance characteristics for the application. These graphs illuminate circumstances within an application where the discovered technology is worth pursuing. After multiple discovery-application matches are evaluated for cost/benefit, a ranked list can be produced. The ranking suggests the best investment of research funding for further development of the new technology.

1.3 Outline of the Reported Study

The objective of this study is to develop a technology transfer methodology which incorporates information about a new technology, potential applications, and a quantitative procedure for evaluating the ranking of the most beneficial use and advantages over existing processes. The emphasis is on "what can be" rather than "what is." The methodology is prototyped in a two-level decision-making system capable of selecting a technology-product/application match, evaluating the design of the product and producing a cost/benefit analysis of the technology transfer process. The final output is a ranked list of new technology-application matches.

1.3.1 "New Technology" Knowledge Base

The knowledge base for new technology is developed for a recent laboratory discovery that shows great promise for tribological surface applications—multilayer diamond thin-film coating. Knowledge about the process capabilities, limitations, surface physics, manufacturing processes involved, cost factors and design parameters is encoded in a production system knowledge base.

Although the contents of the new technology knowledge base are domain specific to the diamond coating process, the underlying knowledge base structure is domain independent. This knowledge base is designed with extensibility in mind.

1.3.2 "Needs and Applications" Knowledge Base

A knowledge base containing potential applications of the new technology is developed. Knowledge about the potential applications is encapsulated in a production system knowledge base. The knowledge base contains current properties, limitations and process

parameters for the application. Other important information included is desired properties, manufacturing processes involved, cost factors and design parameters. A skeleton of this knowledge base is provided in Table 1-1.

Table 1-1. Potential applications knowledge base

Product Domain	Specific Product Application	Desirable Surface Properties
High Sliding, Concentrated Contacts	gearing	high K, ρ, C high hardness low $E\alpha\Delta T$ low surface roughness high thermal shock resistance large lubricating film thickness
Bearings	rolling element • isothermal/Newtonian regime • high temperature • corrosive environment	low peak coefficient of friction low ΔT
Brakes	frictional	high K, ρ, C low $E\alpha\Delta T$ high thermal shock resistance high coefficient of friction stable coefficient of friction high hardness long life
Clutches	frictional	

The needs and potential applications knowledge base is similar to the new technology knowledge base structure in that it is domain specific to the various application areas described above, but the underlying structure is domain independent. This knowledge base is designed with extensibility in mind. Knowledge regarding new potential application domains can be added as it is acquired.

1.3.3 Inference Method for Matching Technology to Needs and Applications

An inference engine provides the capabilities for matching the new technology to potential applications. A quantitative procedure for ranking application alternatives is developed. Ranking is important because high payoff applications should be attacked first.

The CLIPS production system expert shell provides a robust pattern-matching capability that was beneficial in developing a reliable matching and ranking methodology [Giar91]. CLIPS has the capability of representing knowledge in the form of facts, rules and procedures. Tew [Tew87] developed a knowledge-based approach to searching an on-line software application database using the EXSYS expert system shell. EXSYS, like CLIPS, is also based on the production rules approach to knowledge representation.

Regardless of the inference engine used to implement the matching methodology, the critical exercise is developing a reliable quantitative method. It is also important to note the separation of the knowledge from its use. As Gonzales and Dankel [Gonz93] point out, it is this separation that provides the true power of knowledge-based systems:

> This separation allows us to develop different applications by having to create only a new knowledge base for each application. The generic reasoning technique (i.e., the inference engine) is not modified. For example, the basic troubleshooting approach used in medical diagnosis is similar to that used by a mechanic in diagnosing a fault in an automobile. Only the domain is different. Once the generic knowledge is encoded, it can be applied to either domain, thereby greatly simplifying the development process (p. 1).

The inferencing process used to match new technology to potential product applications is based on a technique developed by Forgy [Forg82]—the Rete algorithm—for efficient many-many pattern matching. The Rete algorithm is integral to the CLIPS system used in developing the knowledge bases for this study.

1.3.4 Algorithmic Methodology

Ultimately, the algorithmic analysis module will be a tightly integrated, interactive modeling system. The prototype system developed for this study relies heavily on flexible engineering mathematical modeling tools and spreadsheet applications. This approach was taken to prove the concept while minimizing the amount of software development. The simulations are complicated by the use of dimensional data. Handling units in traditional, from-the-ground-up software development projects requires a great deal of effort that could be channeled towards larger goals. The engineering modeling tool used in

building the simulations provided a powerful, built-in units-handler that proved indispensable.

The spreadsheet approach does have its drawbacks. Among them is a loose system integration which places a burden on the user to make sure that the proper data gets used by the right application at the right time. Results are also slow to appear due to increased interactive demands on the user. A tightly integrated analysis system would solve many of these problems.

1.3.5 Cost/benefit Modeling

Cost/benefit analysis is used in this study to attach costs to performance gains. The components of the cost figure may include yearly production volumes, unit costs, interest rates and time on the market for the product. Rej and Alexander [Rej94] developed a comprehensive cost model for a plasma source ion implantation (PSII) facility—a process similar to that used in applying diamond. The best pricing and marketing possibilities for the new technology are explored relative to the cost of development and cost of full scale production. Relevant data for this segment of the study are compiled in the references.

1.4 Considered Example for Laboratory Process Technology Transfer

The diamond coating process currently under development at the University of Florida is used as the example technology transfer process. Applications to tribological systems are investigated as sample products. A cost/benefit analysis for the research and development effort is performed. A life-cycle cost comparison illustrates the advantages and disadvantages of these new application of technology versus traditional approaches.

One reason for this selection is the great potential for this process to generate considerable savings to the U.S. economy. A 1981 study by the ASME Research Committee on Lubrication for the U.S. Department of Commerce [ASME81] estimated that the U.S. economy loses $60 billion per year due to friction and surface wear.

1.4.1 Multilayer Diamond Thin Film Coating

The process examined in this study for product applications is in early stages of development. In essence, the process involves creating a very uniform surface roughness on metallic surfaces using a laser. Since diamond has such a small thermal expansion coefficient, it is difficult to grow on metallic surfaces where the thermal mismatch is high. The laser process allows for a larger contact surface area between the diamond film and the metallic surface. The larger contact surface area is intended to grade the residual stresses across the diamond film—allowing the film to better adhere to the surface. Since carbon from the diamond tends to diffuse into the surface of iron-based substrates, it is desirable to introduce a buffer layer between the diamond film and the substrate. For the purposes of this study, it is assumed that effective bonding between the diamond film and the substrate is realizable. Patents have been recently submitted on the laser surface preparation process. There is no published information in the literature describing the process.

1.4.2 Tribological considerations

Tribological considerations for multilayered surface treatments include heat partition and transient temperature effects, friction factors for high slide-to-roll ratios, effect of lubricant properties on temperature and wear in sliding concentrated contacts, wear resistance, elastohydrodynamics and thermohydrodynamic shear zone thickness.

1.5 Preview of Subsequent Chapters

Chapter 2 reviews the pertinent literature for this multidisciplinary study. Chapter 3 examines the thermal properties of diamond coated substrates. The thermal expansion mismatch and the formation of graphite layers when coating steel with diamond necessitates the use of a buffer layer. A procedure is developed for determining the transient heat transfer properties of multilayered substrates. Chapter 3 also develops a method for approximate evaluation of the nominal thermal stresses and predicting the fatigue life of diamond coated steel. Process attributes for the diamond coating discovery are examined in Chapter 4. Cost modeling for the coating process is also developed in Chapter 4.

Potential applications are investigated in Chapter 5. The case illustrations are for different gear applications. The qualitative matching process is developed in Chapter 6. This methodology relies heavily on the analytical methods presented in Chapters 3, 4 and 5, and ties together the process attributes presented in Chapter 4 and the potential applications of Chapter 5. In Chapter 7, a step-by-step procedure is presented for building the qualitative matching knowledge base structure and for executing a sample matching process. Chapter 8 provides a step-by-step procedure for taking the results of the qualitative matching process and performing necessary quantitative analyses to determine a ranking of potential applications. Chapter 9 summarizes the findings and the conclusions of this study, and presents suggestions for further research.

CHAPTER 2
LITERATURE REVIEW

The problem under consideration incorporates information from a variety of disciplines. These include technology transfer, coating technology, Tribology and the technology of machine elements whose design can be constrained by surface temperature—such as gears, bearings and clutches/brakes.

Also relevant to this investigation are the disciplines of knowledge-based systems, attribute matching and software engineering.

2.1 Technology Transfer

Technology transfer from the laboratory to industrial/commercial implementation has recently received a large amount of public attention. Many state-assisted consortiums, such as Enterprise Florida, are actively soliciting researchers ready to commercialize their research results. The shrinking Department of Defense budget has forced many contractors, once solely dependent on the government for funding, to turn to civilian applications of their technology. Thus technology transfer is of a great interest to both the individual researcher or small research team and the established high-technology manufacturing base.

The published literature has many examples of the philosophical issues related to technology transfer. Among these issues are how to best manage the process, how to best disseminate new ideas for possible transfer and how to best conduct the communication process between the technology provider and the technology user/implementor. Other commonly explored issues include the decision making process on choosing between competing technologies for a particular application. There are few data on the analytical aspects involved in evaluating potential applications for new discoveries.

Gibson and Niwa [Gibs91] proposed a new research area in "knowledge-based technology transfer," (p. 503). Their study models and integrates research and theory in the areas of knowledge-based systems and technology transfer. The researchers found that knowledge-based systems and technology transfer share a common objective: effective knowledge transfer. In their reported work, they discuss three historically linear technology transfer models based on the knowledge–consulting paradigm. These models all disseminate technological knowledge in one direction—from technology expert to technology user. The first of these models, the Appropriability Model, was in use during the late 1940s through the 1950s. This model was based on "thinking good thoughts" and publishing results, with the belief that good technologies would sell themselves. The Dissemination Model, in use during the 1960s and 1970s, emphasized diffusion of information transfer of expert knowledge to users, opinion leaders or willing receptors. The philosophy behind this model is that once the communication linkages are established, the knowledge will flow. The third and most modern approach, the Knowledge Utilization Model, emphasized the behavioral issues such as interpersonal communication between technology researchers and technology receptors, and overcoming organizational barriers and finding technology transfer facilitators.

The researchers proposed a new two-way approach, the Communication-Based Model which enhances the exchange of knowledge from the technology provider and requirements from the technology user. In this model, technology transfer is a continuous, interactive process where idea exchange between provider and user is simultaneous and continuous until "convergence of knowledge developers, and users in terms of technology development, acceptance and application," ([Gibs91] p. 504). The Communication-Based Model uses a knowledge-sharing methodology where communication is between expert user and expert user. In this way knowledge is decentralized and transferred in both directions.

Sage [Sage89] reported a systems engineering technology management process methodology supporting the research and development of emerging technologies. The study presents issues related to the identification of new applications and the evaluation of their potential. Sage emphasizes the dynamics of market push and pull and says "successful innovation more frequently is driven by market pull than by technology push," (p. 311). This report offers a framework for assessing technology potential for further development from a managerial and societal perspective but does not propose an analytical framework.

An expert system technology evaluation tool developed by Siemens AG was reported by Reminger [Remi91]. Siemens uses this tool to assess the viability of new technology for further development. Over 1000 rules were implemented in the system to capture the technology evaluation model. The tool diminishes the time that technology evaluation and planning requires. The example cited was the evaluation of 11 technologies in 2 weeks versus the pre-tool time of one year. The compressed time factor is a source of competitive advantage for Siemens. No details were provided concerning the system's evaluation strategy.

Luxhoj [Luxh94] compared alternative technology transition strategies using the Analytic Hierarchy Process (AHP) "as a justification methodology to integrate and evaluate both quantitative and qualitative factors in this complex, multi-attribute problem domain," (p. 81). The problem domain studied was the Federal Aviation Administration's plan to modernize its Air Traffic Control Towers with advanced technology flight management systems. Luxhoj concluded that AHP may be successfully applied to evaluate multi-attribute problems which consist of both quantitative and nonquantitative influence factors. AHP was developed by Saaty [Saat90] to provide a systematic way to make decisions in complex, multi-factor situations where decision parameters are interdependent. Many deficiencies have been reported for AHP and alternative decision support techniques have been proposed [Ra91].

2.2 Coating Technology

The properties of diamond films and other synthetic diamond materials are collected in several references. Among the most comprehensive are properties compiled by Field [Fiel92] and Davis [Davi93]. The work summarized by Field [Fiel92] includes complete property information in the appendix which is divided into general properties, mechanical properties, thermal properties, optical and electrical properties, and industrial products. Yoder [Yode93] summarizes mechanical and elastic properties of diamond and compares these properties to common engineering materials. The study also includes a look at high temperature and tribological applications of diamond.

Lux and Haubner [Lux93] discuss various low pressure methodologies for producing diamond films. They also present some simple cost models for producing diamond-coated cemented carbide cutting tool inserts using in situ chemical vapor deposition (CVD). The wear and cutting applications examined in this study rely mainly on the superhard properties of diamond. The researchers conclude their report with a look at short- and long-term outlooks for scaling up production of diamond coated substrates and free-standing diamond sheets. Chief among these concerns is quality assurance for diamond films and multi-layer approaches that will allow different grades of diamond surface to be produced.

Plano and Pinneo [Plan93] discuss goals of CVD diamond research approaches for diamond property enhancement and future nonelectronic diamond applications. The applications discussed include the following:

1. mechanical components—such as machine tools, paper mill paper rollers and ceramic roller bearings;
2. thermal components—such as heat spreaders for high power density devices, packing substrates for high speed microprocessors and gas turbine combustion chamber liners;
3. optical components—such as x-ray deflector windows, liquid-cooled high power laser mirrors and as a laser host material.

The researchers include both thermal and superhard properties of diamond films in their potential application assessment.

Rej and Alexander [Rej94] developed a semiempirical model for the cost of a commercial plasma source ion implantation (PSII) facility. The cost model estimates amortized capital and operating expenses as functions of the surface area throughput T. The model predicts a reasonably sized PSII facility should be able to treat a surface area of 10^4 m^2 per year at a cost of \$0.01 per cm^2. Many of the same power and capital equipment requirements are similar for CVD and PSII. The PSII facility cost model may be a useful reference for full scale CVD production of diamond films.

2.3 Tribology

Tribology is "the branch of science and technology concerned with interacting surfaces in relative motion and with associated matters (as friction, wear, lubrication, and the design of bearings)," ([Jost91], p. 129). The published literature contains many papers relevant to this study. The main area of importance focuses on sliding concentrated contacts and the associated heat generation.

O'Donoghue and Cameron [O'Do66] studied friction and temperature in rolling sliding contacts and were able to develop a correlation for the coefficient of friction based on load, speed, viscosity radius and surface finish of the contacting surfaces. Seireg and Hsue Seir81 investigated the effects of lubricant properties on temperature and wear in sliding concentrated contacts. The researchers found that viscosity does not appear to be the significant property of the lubricant temperature rise and wear rate. Li and Seireg [Li89] developed a dimensionless empirical formula for calculating the coefficient of friction in sliding-rolling steel on steel contacts operating in the thermal regime. The formula provides a unifying relationship for all the published data. Othman and Seireg [Othm89] devised an empirical procedure for evaluating the frictional properties in Hertzian contacts subjected to sinusoidal sliding motion. The researchers found "that the friction-velocity function can be adequately approximated by an exponential function in the considered case where the parameters of the function can be readily determined from a multivariate search which minimizes the errors of the peak response at resonant frequencies," (p. 4).

The Rashid and Seireg [Rash86] study of heat partition and transient temperature distribution in layered concentrated contacts is the most relevant tribological study for this dissertation. The researchers developed dimensionless, empirical relationships for predicting maximum temperature rise in layered concentrated contacts. This paper is the basis for the design relationships developed in this dissertation. Chapter 3 extends the methodology presented in this paper to handle multilayered concentrated contacts.

2.4 Machine Elements

The case study in this dissertation compares traditional machine element applications with diamond coated machine elements. General references for machine element design include Shigley and Mitchell [Shig83] and Orthwein [Orth90]. A specific reference which integrates thermal considerations in the design of gear pairs is found in Lin and Seireg [Lin85]. This paper presents a computerized optimization algorithm for the design of gear systems which selects the design parameters to provide the best balance between the meshing elements for strength, surface durability, wear resistance and surface temperature. Formulas presented in this paper are central to the diamond coated gear system design presented in this dissertation.

2.5 Knowledge-Based Systems

Knowledge-based systems are playing an increasingly important role in the design of engineered products. Knowledge-based systems come in many flavors, including automated reasoning machines, neural networks, production systems, blackboard architectures and expert systems. Artificial Intelligence is a general field that loosely contains all of these special forms of knowledge representation and management. Winston [Wins92] and Rich and Knight [Rich91] are references of general applicability to the field of Artificial Intelligence. These texts contain a thorough overview of many methodologies that comprise the field. Expert systems are of more specific interest to this work. Gonzales and Dankel [Gonz93] provide a good introduction to a variety of expert systems approaches and include information on societal, ethical and legal ramifications of their use. CLIPS

[Giar91, Giar93] is an object-oriented, rule-based, expert system language developed by NASA. This dissertation research takes advantage of many of the features in CLIPS to model the technology transfer methodology.

2.6 Attribute Matching

Critical to the success of selecting product applications for laboratory discoveries is the ability to effectively and efficiently match attributes common to both. In this case, a set of rules determines how applications will match discoveries. The rules pattern match on facts or objects and take appropriate action. Matching single attributes in the pattern of a rule is trivial. More realistic situations involve multiple rules with multi-attribute patterns. In this case, a many–many match algorithm coupled with a conflict resolution strategy is employed to handle the exponential number of match permutations possible.

One efficient many–many match algorithm is Rete[1] which was developed by Forgy [Forg82]. Rete gains efficiency from three major sources (see Rich and Knight [Rich91]):

1. the temporal nature of data;
2. structural similarity in rules;
3. persistence in variable binding.

The Rete works by maintaining a network representation of rule conditions. As rules fire (match patterns and take action), the state description changes. Some new rules will now apply and some old ones may no longer apply. Rete only checks those rules effected by the change in state and thus saves recomputing the entire matching network after each rule firing. The Rete algorithm is used by the CLIPS system for its matching strategy.

2.7 Software Engineering

Software Engineering is a relatively new field in engineering. It deals with the entire life-cycle of software development: need recognition, requirements gathering, information

---

1. *rete*: an obscure synonym for net

modeling, algorithm development, software system design, verification and validation, documentation and maintenance. This field has gained increasing importance as software systems have grown increasingly complex. Sommerville [Somm92] provides a comprehensive reference that provides an excellent introduction to the field.

Object-Oriented Design (OOD) is a modern software design paradigm that is ideal for modeling complex software systems. An excellent reference on the subject is the text by Booch [Booc91]. Booch provides an overview of OOD and incorporates five complete examples of systems developed using this technique.

CHAPTER 3
THE THERMAL PROBLEM

To predict the behavior of diamond coated components, it is necessary to model the thermal process in sliding contacts. Many tribological applications involve Hertzian contacts with high slide-to-roll ratios. The goal of the transient heat transfer analysis developed in this chapter is to determine the maximum temperature rise, thermal gradients, and thus maximum thermal stress in layered, sliding concentrated contacts. In this chapter, computationally efficient, simplified, predictive equations are developed for the maximum temperature rise in diamond coated substrates.

The temperature gradient in the substrate causes a surface stress. Simplified thermal stress equations are developed in this chapter to determine the magnitude of the surface stress. These values are used in an empirical equation to determine a life debit due to the thermal stress.

3.1 Transient Heat Transfer

Closed form solutions for transient heat transfer problems are available for only a few simple geometries and boundary conditions. Superposition can be used to combine boundary conditions to yield new solutions in some cases. As the number of boundary conditions increases and the geometric part complexity increases, the use of superposition becomes unwieldy. Computer modeling is the most practical way to predict the thermal response in these cases. Finite element methods and finite difference methods are two widely used approaches for determining temperature distribution in transient heat transfer cases. These methods are very flexible and are commonly used to solve a variety of transient heat transfer problems.

Although the expense of running finite element and finite difference models has been steadily decreasing as the power of computer workstations has increased, structuring these approaches for optimization problems results in computationally expensive efforts. For design purposes it would be useful to have a computationally efficient method available—even if it sacrifices accuracy to gain insight into interaction among the problem's parameters. For the purposes of this study, the maximum temperature rise due to transient effects is more important than the thermal distribution. The following sections describe a closed form approach for predicting the maximum temperature rise for a few special cases relevant to diamond coated substrates.

3.1.1 A Heat Source With a Hertzian Distribution Moving Over a Layered Semi-Infinite Solid

Rashid and Seireg [Rash86] developed dimensionless relationships for heat partition and transient temperature distribution in layered concentrated contacts. The relationships are based on curve fits of maximum temperature rise in transient heat transfer cases. The data were generated from three dimensional finite difference models for a variety of geometries and boundary conditions. The computer simulation was able to repeat the analytical results of Blok [Blok37] and Jaeger [Jaeg42] for the simple case of a heat source moving over a semi-infinite solid.

3.1.1.1 Heat source moving over a semi-infinite solid

The case of a heat source moving over a semi-infinite solid was used to check the accuracy of the computer simulation developed by Rashid and Seireg. Blok and Jaeger obtained the following relationship for the maximum rise in surface temperature via a series approximation:

$$T_s - T_B = 1.123\frac{q_t}{l}\sqrt{\frac{l}{K\rho CU}}. \tag{3.1}$$

For the same conditions, Rashid and Seireg obtained the following dimensionless result using a finite difference simulation:

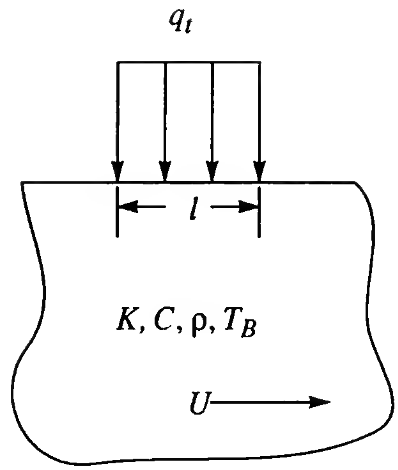


Figure 3-1. Moving semi-infinite solid under a stationary heat source.

$$\frac{(T_s - T_B)K}{q_t} = 1.03\left(\frac{\rho C U l}{K}\right)^{-0.5} . \tag{3.2}$$

Equations (3.1) and (3.2) are in general agreement with the differences in the constants attributable to numerical approximation in the computer model. See Figure 3-1.

3.1.1.2 Heat source with a Hertzian distribution moving over layered semi-infinite solid

This case examines a single-layered substrate with a moving heat source of Hertzian distribution. This heat distribution is equivalent to the heat generated in a contact problem such as a semi-infinite cylinder rolling and sliding over a semi-infinite planar surface. See Figure 3-2. Rashid and Seireg obtained the following relationships:

$$\frac{(T_s - T_{so})K_o}{q_t} = 1.137\left(\frac{D}{l}\right)^{1.4}\left(\frac{h_o}{l}\right)^{0.47}\left(\frac{l_e}{h_o}\right)^{0.78} \tag{3.3}$$

$$\frac{(T_o - T_s)K_o}{q_t} = 1.164\left(\frac{l}{D}\right)^{0.026}\left(\frac{K}{K_o}\right)^{0.026}\left(\frac{h_o}{l}\right)e^{-180\times10^{-6}\frac{l_e}{h_o}} \tag{3.4}$$

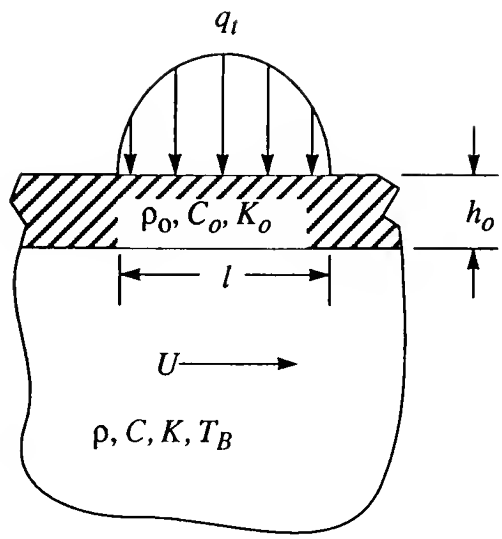


Figure 3-2. Layered semi-infinite solid moving under a stationary heat source with a Hertzian distribution.

$$\frac{T_S K}{q_t} = 1.12\sqrt{\frac{K}{\rho C U l}} \tag{3.5}$$

$$D = \sqrt{\frac{5Kl}{\rho C U}} \tag{3.6}$$

$$l_e = \frac{1}{5}\frac{U h_o^2 \rho_o C_o}{K_o} \tag{3.7}$$

where

D = the temperature penetration depth at the trailing edge

l_e = the required entry distance for temperature penetration across the film

T_{so} = the maximum rise in the solid surface temperature for a layered semi-infinite solid

T_s = the maximum rise in the solid surface temperature for the unlayered semi-infinite solid with the same heat input.

T_o = the maximum rise in the layer surface temperature.

3.1.1.3 Illustrative examples

Two examples demonstrate the effects of a conductive layer and an insulative layer on a stainless steel substrate. Parameters for these cases are set to the following:

$$q_t = 100 \text{ Watt/mm}$$
$$U = 13700 \text{ mm/sec}$$
$$l = 1.375 \text{ mm}$$
$$h_o = 0 \text{ to } 16 \text{ μm.}$$

The stainless steel is AISI 304. The temperature rise in the stainless steel is denoted by T_{ss}.

Conductive layer on stainless steel. The conductive layer applied to the AISI 304 substrate is diamond. The temperature rise in the diamond is denoted by T_d. T_{ss} and T_d are plotted versus diamond layer thickness in Figure 3-3. The temperature rise in the diamond shows a slight increase as the layer thickness increases. The temperature rise in the stainless steel decreases approximately 50°C for a 16μm layer thickness.

Insulative layer on stainless steel. The insulative layer applied to the AISI 304 substrate is silicon nitride (Si_3N_4). The temperature rise in the silicon nitride is denoted by T_{if}. T_{ss} and T_{if} are plotted versus silicon nitride layer thickness in Figure 3-4. The temperature rise in the silicon nitride shows a dramatic increase as the layer thickness increases. The temperature rise in the stainless steel decreases approximately 200°C for a 16μm layer thickness.

3.1.2 Extension to Multi-Layered Semi-Infinite Solid

The single-layer approach is not adequate for modeling diamond coating on substrates such as steel. In this case it is necessary to introduce a buffer layer to prevent the formation of graphite at the diamond/steel interface. The goal here is to extend the single-layer to two layers without repeating the extensive finite difference modeling approach used by Rashid and Seireg. An approximate method is presented here and is intended for use as a

Case: U=12700mm/sec, l=0.250mm

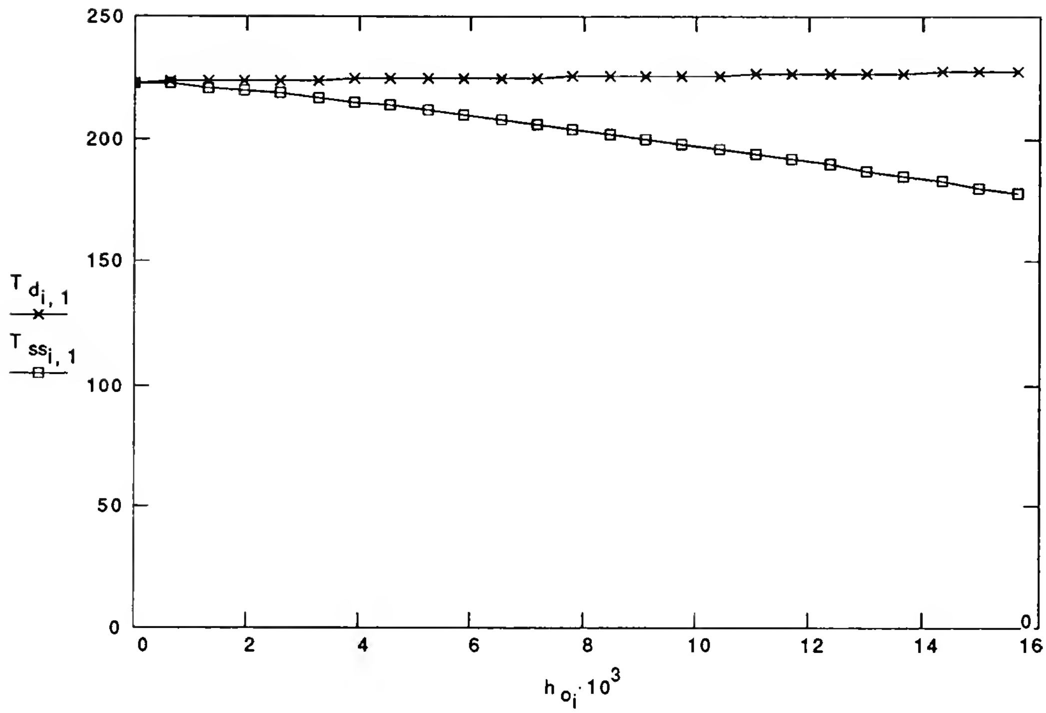


Figure 3-3. Maximum temperature rise (°C) in AISI 304 substrate (T_{ss}) and diamond surface layer (T_d) moving under a stationary heat source with a Hertzian distribution.

design approximation. The results should be verified via a finite element model. Figure 3-5 illustrates the two-layer transient heat transfer problem.

The following dimensionless relations were developed for computational convenience:

$$\theta_{sso} = \frac{(T_s - T_{so}) K_o}{q_t} \left(\frac{K_d}{K_o}\right) \tag{3.8}$$

$$\theta_{os} = \frac{(T_o - T_s) K_o}{q_t} \left(\frac{K_d}{K_o}\right) \tag{3.9}$$

$$\theta_s = \frac{T_s K}{q_t} \left(\frac{K_o}{K}\right) \left(\frac{K_d}{K_o}\right) \tag{3.10}$$

Case: U=12700mm/sec, l=0.250mm

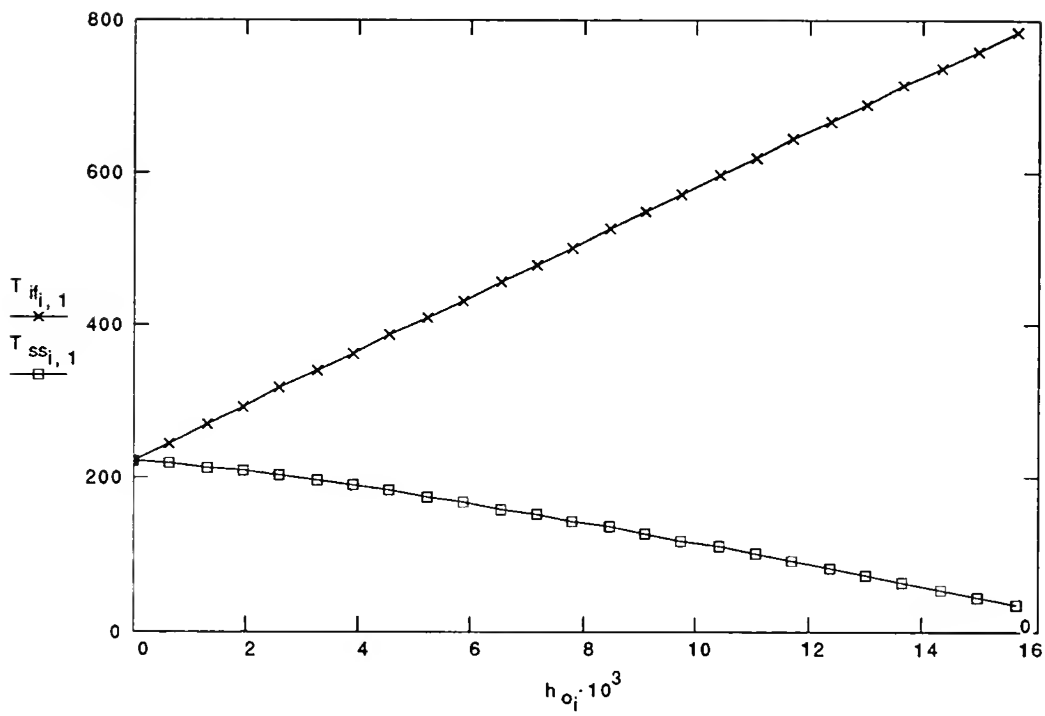


Figure 3-4. Maximum temperature rise (°C) in AISI 304 substrate (T_{SS}) and silicon nitride surface layer (T_{if}) moving under a stationary heat source with a Hertzian distribution.

$$\theta_o = T_o\left(\frac{K_d}{q_t}\right) = \theta_{os} + \theta_s \tag{3.11}$$

$$\theta_{so} = T_{so}\left(\frac{K_d}{q_t}\right) = \theta_s - \theta_{sso} \tag{3.12}$$

where

θ_{sso} = the dimensionless maximum temperature rise in the solid surface relative to the maximum unlayered surface temperature rise

θ_{os} = the dimensionless maximum temperature rise in the surface layer relative to the maximum unlayered surface temperature rise

θ_s = the maximum unlayered surface temperature rise

θ_{so} = the dimensionless maximum temperature rise in the solid surface

θ_o = the dimensionless maximum temperature rise in the surface layer.

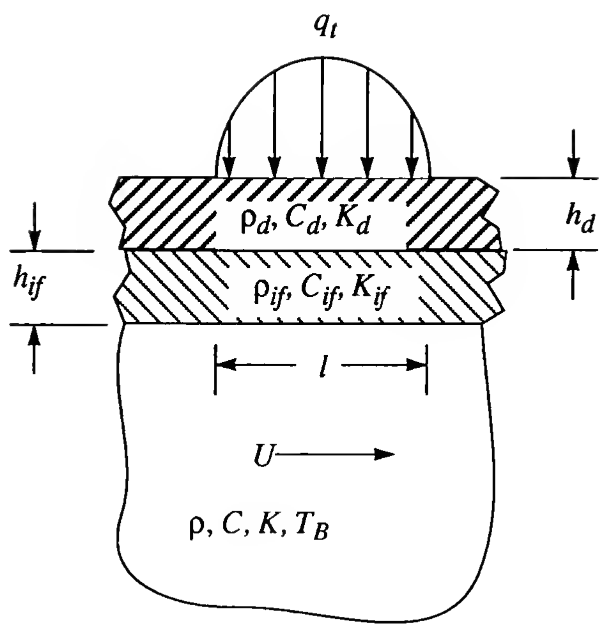


Figure 3-5. Two-layered semi-infinite solid moving under a stationary heat source with a Hertzian distribution.

3.1.2.1 Procedure to predict multilayer temperature rise

A numerical example illustrates the simplified procedure used to predict maximum temperature rise in a multilayered solid. The procedure involves the following steps:

1. For a given velocity, U, and length of contact, l, calculate the dimensionless temperature rise, θ_{o_d} and θ_{so_if}, for the case of a diamond layer on the buffer layer substrate using equations (3.11) and (3.12). The values for the surface layer thickness, h_d, should vary from 0 to a specified upper limit (0.016 mm was used in this study).

2. The ratio between the surface layer and buffer layer maximum temperature rise is now determined. The ratio will remain constant, although the temperature values will be adjusted up or down based on the uncoated steel temperature.

3. For the same U and l, calculate the dimensionless temperature rise, θ_{o_if} and θ_{so_ss}, for the case of a buffer layer on the steel substrate using equations (3.11) and (3.12). For simplicity, the same values for h_{if} are used as for h_d in step 1 above. See Figure 3-6.

4. The maximum dimensionless temperature rise in the buffer layer, θ_{if}, in the steel substrate, θ_{ss}, and in the diamond layer, θ_d can now be determined. First, the steel substrate temperature rise is scaled by the ratio of the buffer layer temperatures. See equation (3.13). A scaling factor, Δ, is then determined in equation (3.14). This factor

Case: U=12700mm/sec, l=0.250mm

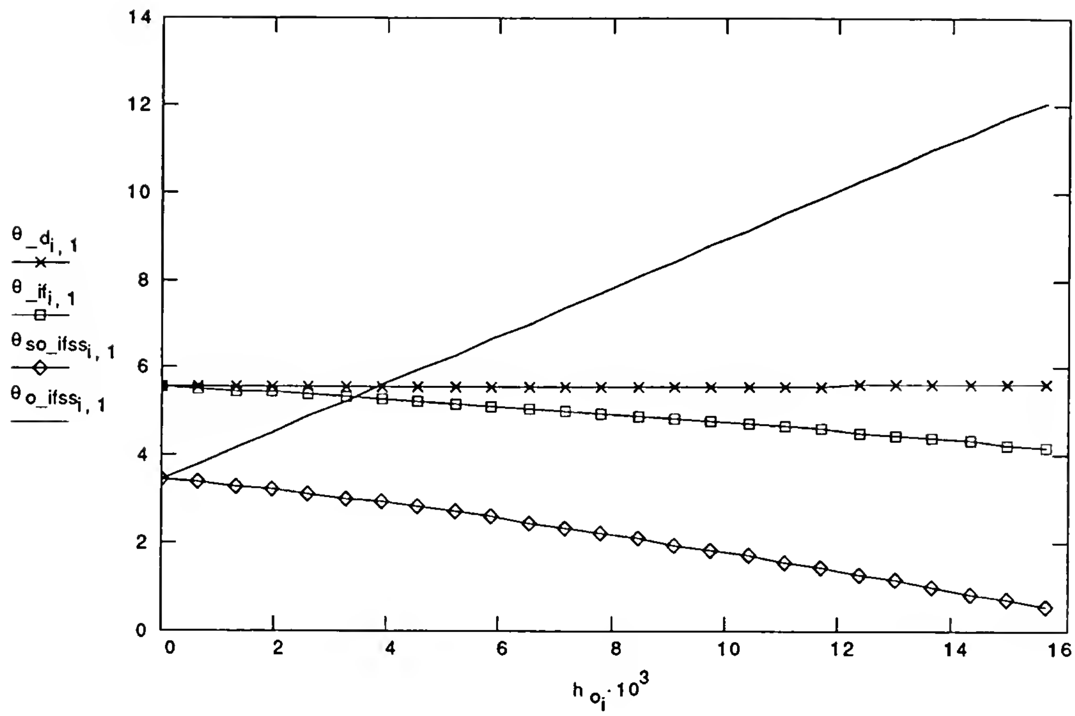


Figure 3-6. Dimensionless temperature rise (prior to data manipulation) in substrate surface and coating layer surface for diamond on AISI 304 stainless steel with a silicon nitride interface layer.

scales the diamond layer, buffer layer and steel substrate temperatures to the uncoated steel substrate temperature, θ_{s_ss}. This factor insures that the temperature rise predictions all start at the uncoated steel substrate temperature at $h_d = h_f = 0$, and then decrease or increase as the coating thicknesses increase. Equation (3.15) shows how Δ is used to calculate the dimensionless temperature rise for the diamond and buffer layers (θ_d and θ_{if}) and the steel substrate (θ_{ss}). See Figure 3-7.

5. The actual temperature rise (T_{ss}, $T_{,if}$, T_d) can now be calculated by equation (3.16). Figure 3-8 shows a typical result for $q_t = 100$ watt/mm^2.

For $i = 1$ to n

$$\theta^i_{ss} = \theta^i_{so\_ss}\left(\frac{\theta^i_{so\_if}}{\theta^i_{o\_if}}\right) \tag{3.13}$$

$$\Delta = \frac{\theta_{s\_ss}}{\theta^1_{ss}}. \tag{3.14}$$

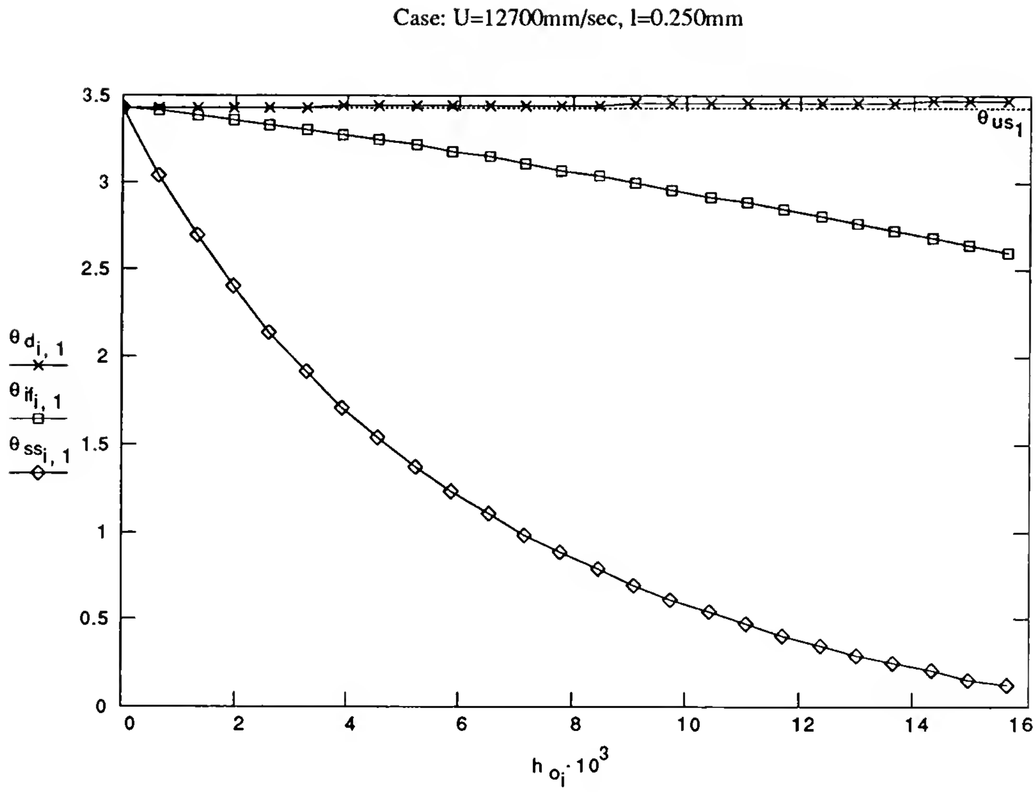


Figure 3-7. Dimensionless temperature rise (following data manipulation) in substrate surface and coating layer surface for diamond on AISI 304 stainless steel with a silicon nitride interface layer.

For $i = 1$ to n

$$\begin{aligned} \theta_{ss}^{i} &= \Delta\theta_{ss}^{i} \\ \theta_{if}^{i} &= \Delta\theta_{so\_if}^{i} \\ \theta_{d}^{i} &= \Delta\theta_{o\_d}^{i} \end{aligned} \tag{3.15}$$

$$T = \theta\frac{q_t}{K_d}. \tag{3.16}$$

3.1.2.2 Effects of varying parameters

The effects of varying the sliding velocity of the solid, U, and the width of contact, l, are examined in this section. Since the temperature rise in an uncoated substrate is inversely proportional to the square root of the sliding velocity, $\Delta T \propto \frac{1}{\sqrt{U}}$, it is expected that the temperature rise for the multilayered case will follow suit. Figure 3-9 illustrates this case. The parameters, except for U, are the same as in Figure 3-8. As predicted, the

Case: U=12700mm/sec, l=0.250mm

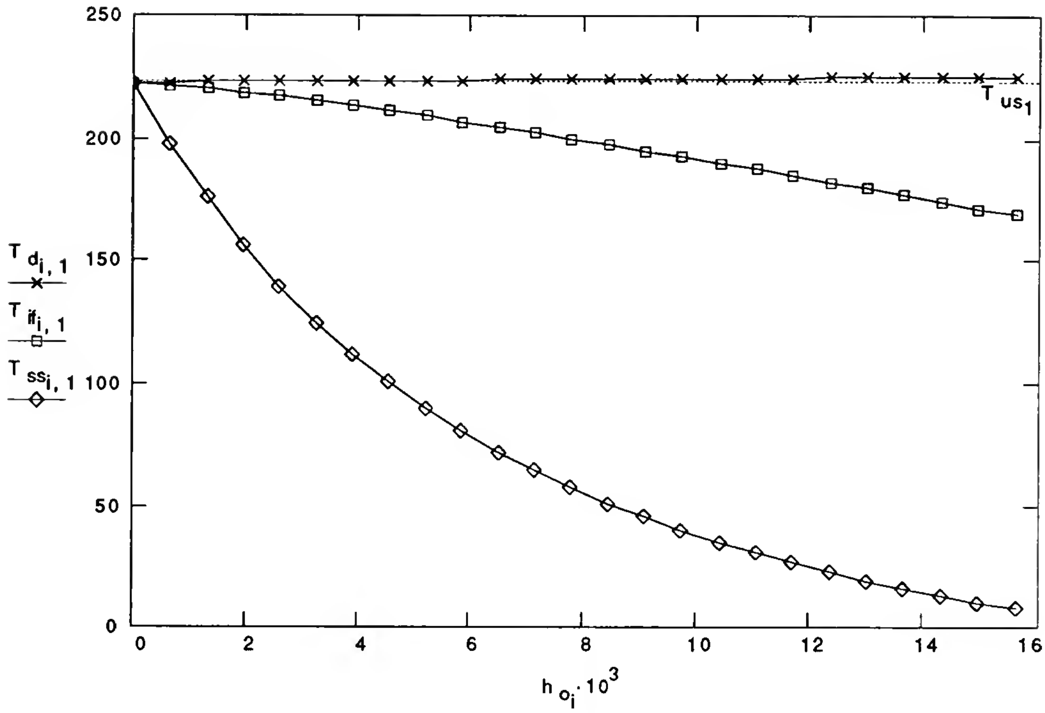


Figure 3-8. Temperature rise (°C) in substrate, interface layer and diamond layer for diamond on AISI 304 stainless steel with a silicon nitride interface layer.

magnitude of the temperature rise for the substrate and layers dropped with the increase in sliding velocity.

The effect of increasing l, the width of contact, is now reviewed. Since l follows the same inverse relationship as U for the unlayered substrate, $\Delta T \propto \frac{1}{\sqrt{l}}$, it is expected that the magnitude of the temperature rise will decrease. Figure 3-10 illustrates this case. The parameters, except for l, are the same as in Figure 3-8. As predicted, the magnitude of the temperature rise for the substrate and layers dropped with the increase in contact width.

3.2 Thermal Stress Considerations

Thermal stresses can have a detrimental effect on the life of a machine component. In this section, simplified equations are developed for predicting the magnitude of the thermal stresses in a multilayered coating. The thermal stress in the substrate will be added to

Case: U=50800mm/sec, l=0.250mm

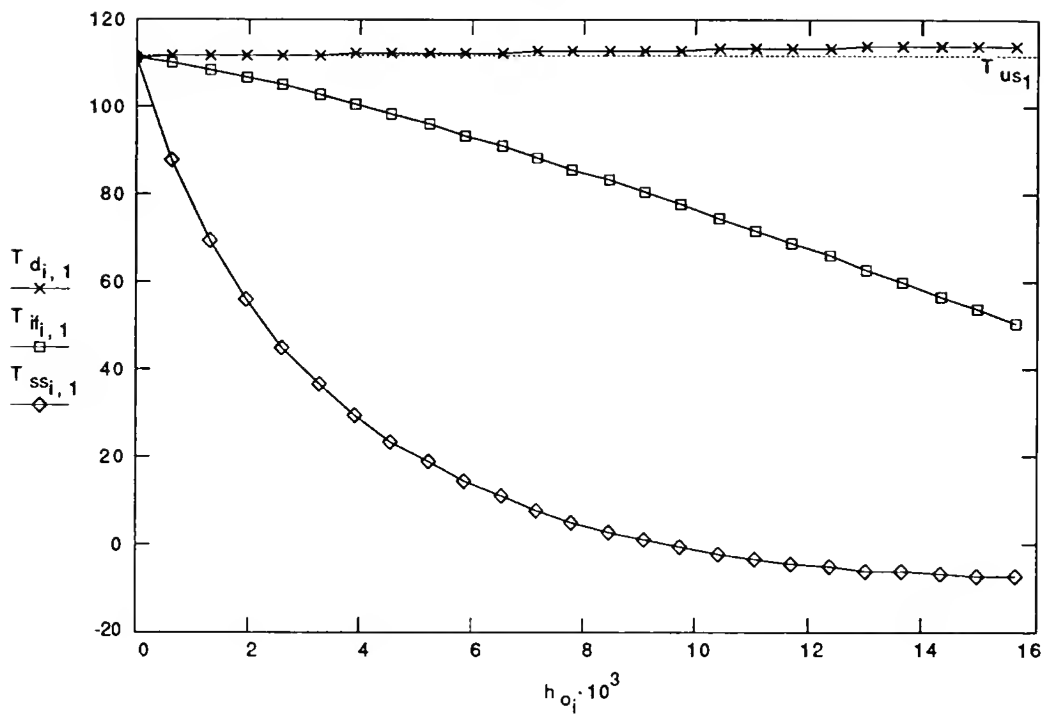


Figure 3-9. Temperature rise (°C) in substrate, interface layer and diamond layer for diamond on AISI 304 stainless steel with a silicon nitride interface layer (velocity increase with length of contact constant).

the contact stress to determine a maximum stress value for calculating a life debit due to thermal fatigue.

3.2.1 Thermal Stress Relationships

Nominal stress relationships for design purposes developed in the following sections refer to the diagram in Figure 3-11. This figure defines the variables used in predicting normal and shear thermal stresses for the case of a multilayer semi-infinite substrate moving under a stationary heat source with a Hertzian distribution.

3.2.1.1 Normal stresses

The normal thermal stress in an axial beam built in at both ends is proportional to the increase in temperature and can be expressed as

Case: U=12700mm/sec, l=1.375mm

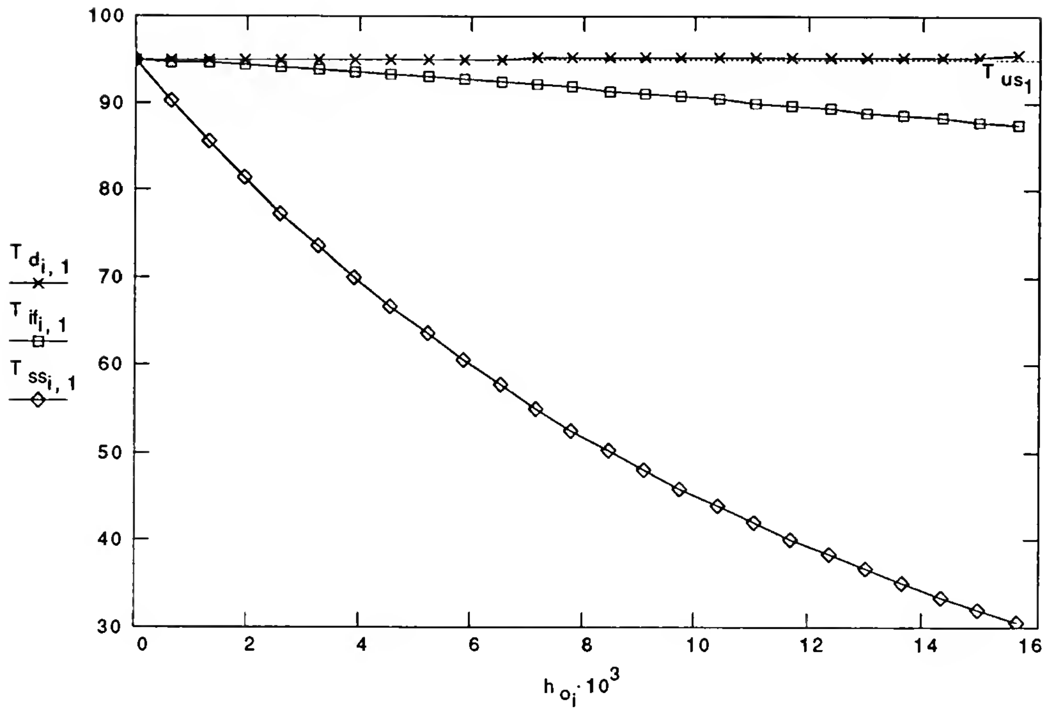


Figure 3-10. Temperature rise (°C) in substrate, interface layer and diamond layer for diamond on AISI 304 stainless steel with a silicon nitride interface layer (velocity constant with length of contact increase).

$$\sigma = E\alpha\Delta T. \tag{3.17}$$

If this equation is used for a model, then we have the following equations to describe the normal stress in the diamond, interface coating and substrate:

$$\begin{aligned} \sigma_d &= E_d\alpha_d T_d \\ \sigma_{if} &= E_{if}\alpha_{if} T_{if} \\ \sigma_{ss} &= E_{ss}\alpha_{ss} T_{ss} \end{aligned} \tag{3.18}$$

where T_d, T_{if} and T_{ss} are the temperature differentials between each of the layers and its substrate. Note that equation (3.18) is a nominal relationship for design approximation only.

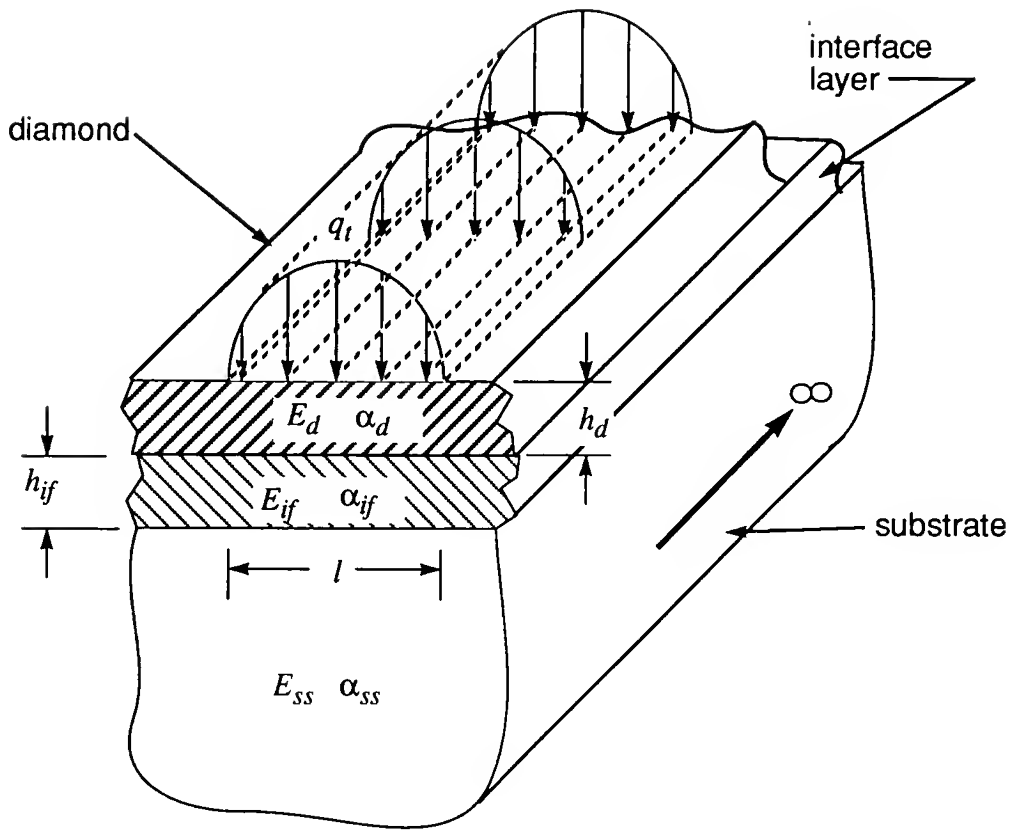


Figure 3-11. Variables used in calculating thermal stress in multilayer coatings.

3.2.1.2 Shear stresses

Shear stress can be determined by dividing the shear force, F_s, by the shear area, A_s. The shear force can be approximated by the difference in normal stresses between two layers times the cross-sectional area or $F_s = (\sigma_2 - \sigma_1) A_c$. The shear stress can now be written as

$$\tau = \frac{F_s}{A_s}$$
$$= \frac{(\sigma_2 - \sigma_1) A_c}{A_s}. \tag{3.19}$$

Now, referring to Figure 3-11, if we substitute $A_c = h\infty$ and $A_s = l\infty$, we have

$$\tau = (\sigma_2' - \sigma_1)\frac{h}{l} \tag{3.20}$$

and we can write the following equations for the shear stress between the diamond and interface layers, and the interface layer and the substrate:

$$\tau_{dif} = (\sigma_{if} - \sigma_d)\frac{h_d}{l}$$
$$\tau_{ifss} = (\sigma_{ss} - \sigma_{if})\frac{h_{if}}{l}. \tag{3.21}$$

3.2.1.3 Example

Figures 3-12 and 3-13 continue with the example from section 3.1.2.

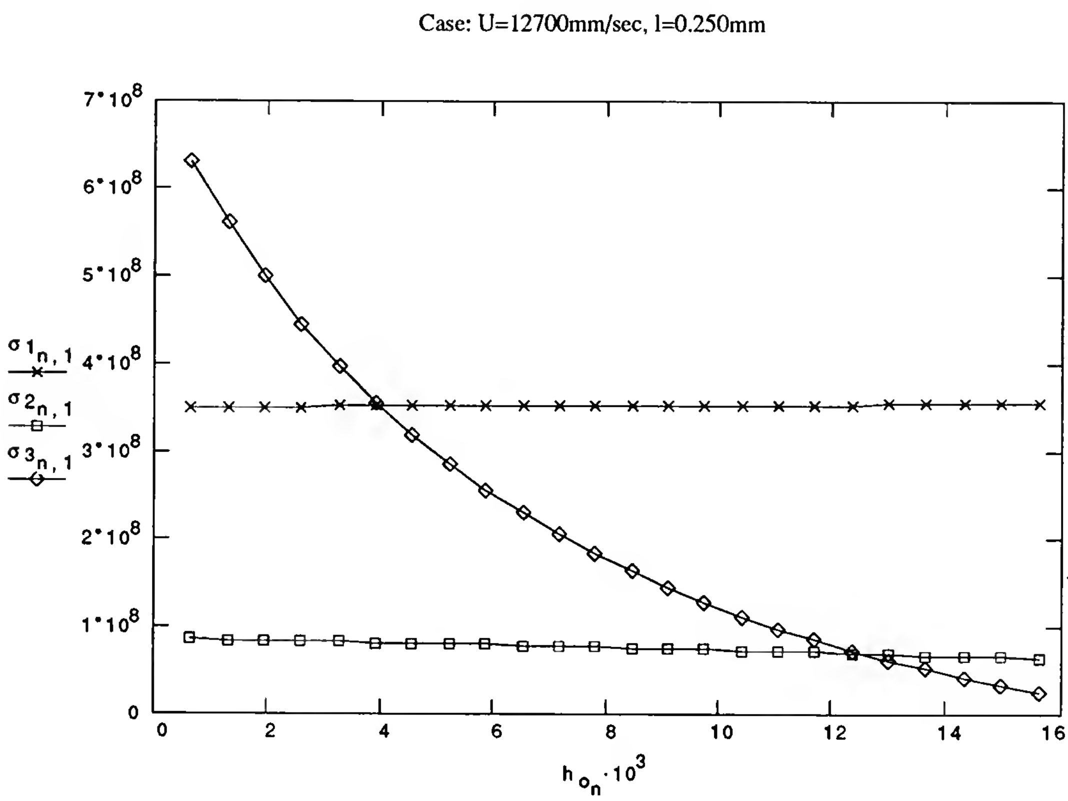


Figure 3-12. Normal stress (Pa) for various thicknesses (mm) of diamond and interface layer. Substrate is AISI 304 stainless steel and interface layer is silicon nitride. Note: $h_{if} = h_d = h_o$, $\sigma_1 = \sigma_d$, $\sigma_2 = \sigma_{if}$ and $\sigma_3 = \sigma_{ss}$.

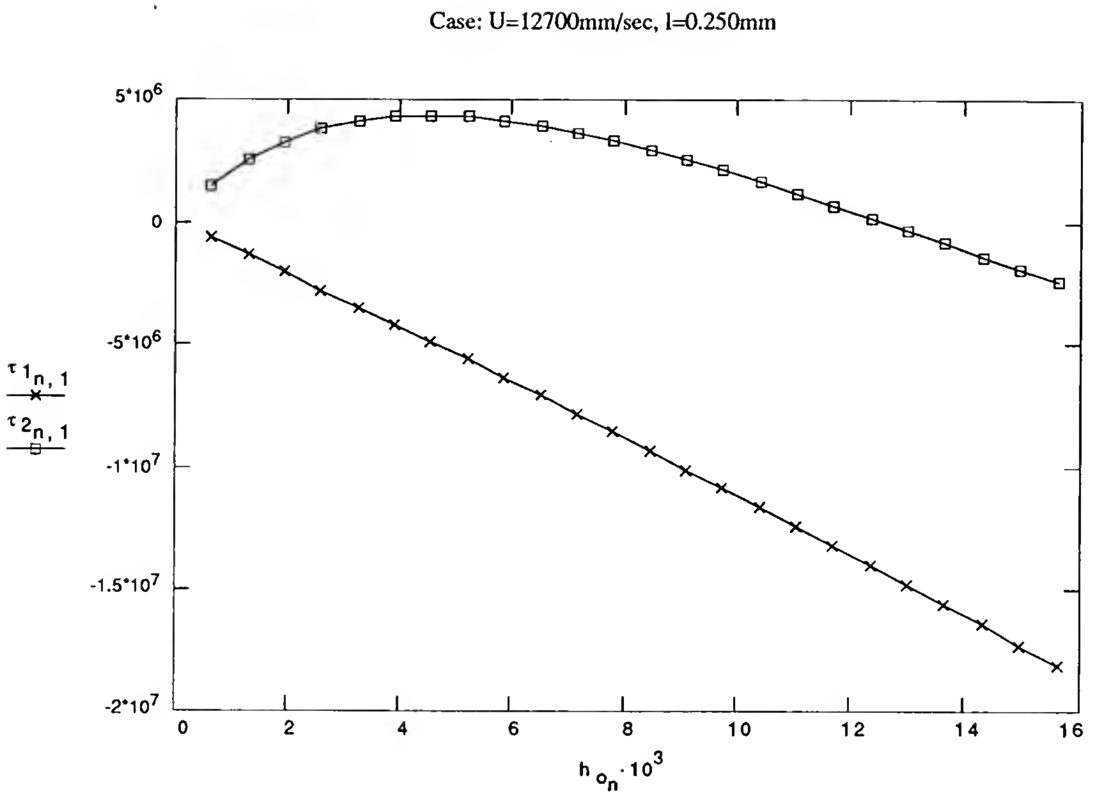


Figure 3-13. Shear stress (Pa) for various thicknesses (mm) of diamond and interface layer. Substrate is AISI 304 stainless steel and interface layer is silicon nitride. Note: $h_{if} = h_d = h_o$, $\tau_1 = \tau_{dif}$ and $\tau_2 = \tau_{ifss}$.

3.2.2 Predicting Fatigue Life Debit Due to Thermal Stresses

Increasing temperature subsequently impacts fatigue life. In this section, the impact on fatigue life due to increasing diamond coating thickness is developed. First a maximum stress value is computed. Next, the stress value is compared to a stress level that represents infinite life.

3.2.2.1 von Mises Stress

Combined rolling and sliding contacts will have stress components due to thermal effects and contact stress. Other components, such as bending stress are ignored. Thermal stresses are considered to be tensile, while contact stresses are compressive. The thermal stress component will therefore be considered σ_{max} and the contact stress component will

be considered σ_{min}. The stresses are combined using the von Mises distortion-energy theory (see Shigley and Mitchell [Shig83]) as follows:

$$\sigma' = \sqrt{\sigma_A^2 - \sigma_A\sigma_B + \sigma_B^2} \tag{3.22}$$

where

σ_A = the maximum normal stress, σ_{max}

σ_B = the minimum normal stress, σ_{min}.

3.2.2.2 Bearing life empirical model

The following empirical relationship is widely used to predict the surface durability of ball bearings:

$$\sigma n^{1/9} = c_l \tag{3.23}$$

where n is the number of cycles and c_l is a constant.

3.2.2.3 Example

Continuing with the previous example, the following additional information is introduced:

1. the level of contact stress, σ_c, is 1000 MPa;
2. $n = 10^7$ cycles is infinite life.

The solution is as follows:

1. $\sigma_{max} = \sigma_{ss}$
2. $\sigma_{min} = \sigma_c = -1000$ MPa
3. calculate σ' and plot (see Figure 3-14)
4. using the minimum value of σ' (should occur at the thickest coating), determine c_l
5. for each value of σ', calculate a new life, n, and plot (see Figure 3-15).

Case: U=12700mm/sec, l=0.250mm

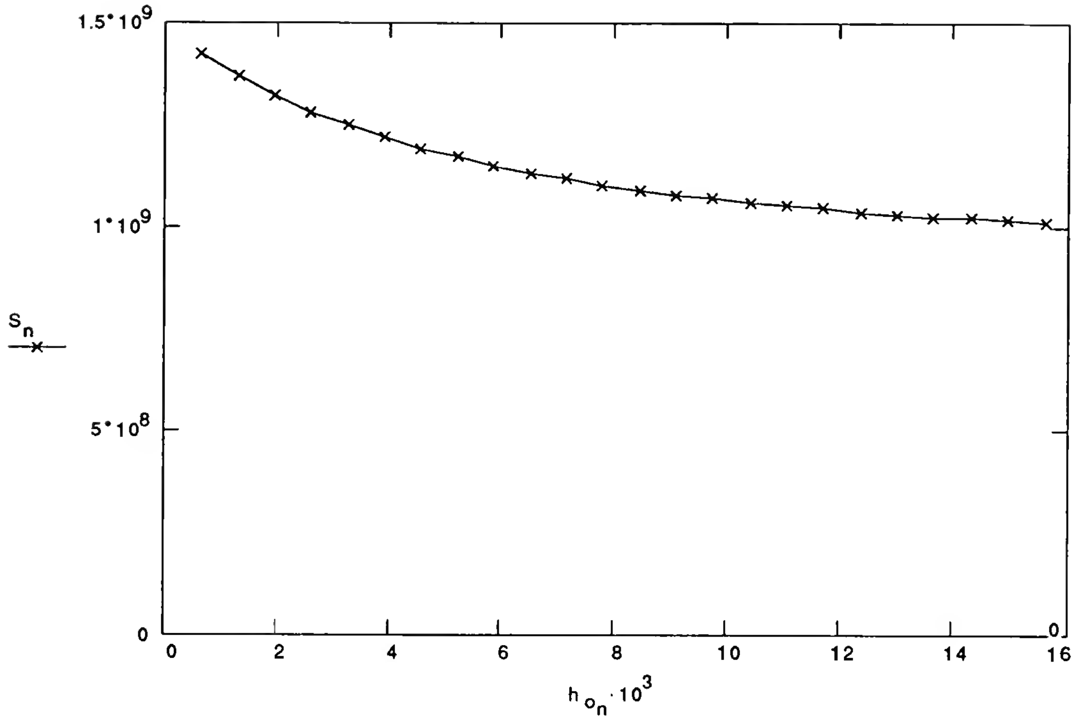


Figure 3-14. von Mises stress (Pa) as a function of coating thickness (mm) in substrate. Substrate is AISI 304 stainless steel and interface layer is silicon nitride. Note: $h_{if} = h_d = h_o$.

It is interesting to note the asymptotic shape to the life curve. It shows clearly that reducing the surface temperature in sliding concentrated contacts drastically increases life.

Case: U=12700mm/sec, l=0.250mm

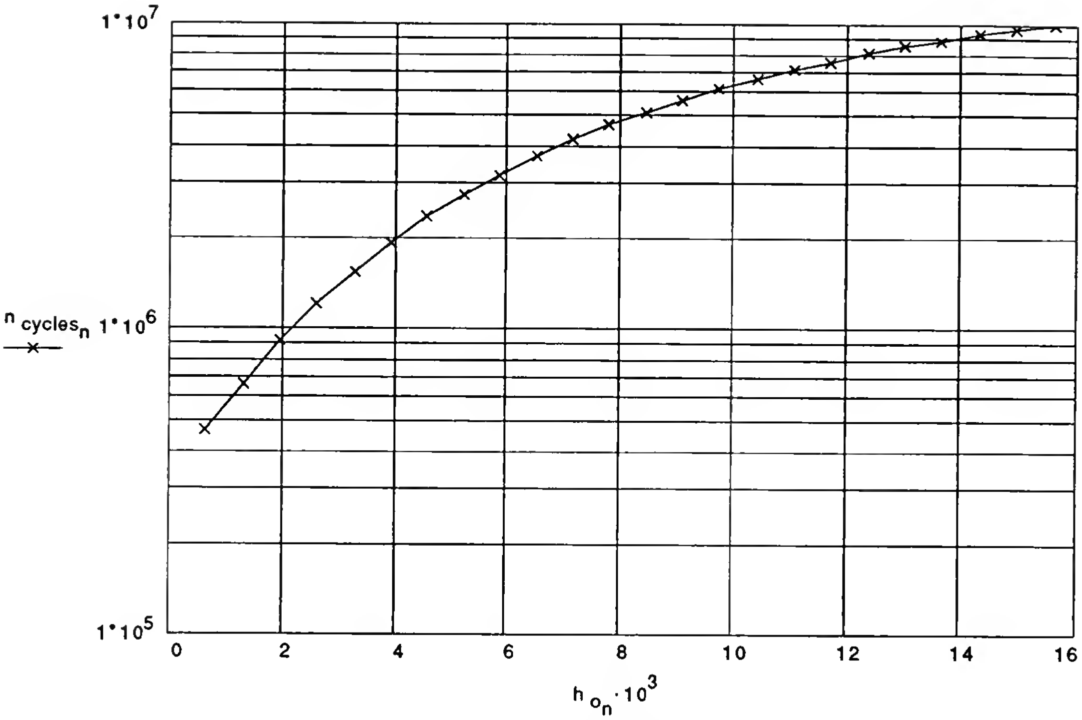


Figure 3-15. Life improvement in cycles versus thickness of diamond film in mm. Substrate is AISI 304 stainless steel and interface layer is silicon nitride. Note: $h_{if} = h_d = h_o$ and contact stress level is 1000 MPa.

CHAPTER 4
PROCESS ATTRIBUTES

This chapter discusses the details of the laboratory discovery used in this dissertation as a case study. The discovery's process attributes are presented from four perspectives. First, the properties of diamonds and diamond films are discussed. Second, the laser surface preparation process under development at the University of Florida is introduced. Third, the chemical vapor deposition process used to grow the diamond film is presented. And finally, the cost aspects of the process are modeled.

4.1 Properties of Diamond

Diamond is an exceptional material. Most of its important properties can be labeled as extreme. It has the highest hardness, the highest thermal conductivity, highest molar density and highest sound velocity of any material known. It also possesses the lowest compressibility and bulk modulus of any known material. The thermal expansion coefficient is also very low and ranks among the lowest of known materials. Diamond is also extremely inert chemically—affected only by certain acids and chemicals that act as oxidizing agents at high temperatures.

In the previous chapter, equations were developed for transient heat transfer in slide-roll scenarios. The heat generated in these cases is due to friction between the sliding surfaces. The temperature rise for the developed cases is inversely proportional to the square root of the product of the surface material's thermal conductivity, specific heat and density, or $\Delta T \propto \frac{1}{\sqrt{K\rho c}}$. Therefore, for transient heat transfer problems where sliding generates the heat, it is crucial to have $K\rho c$ as large as possible in order to minimize thermal effects.

Table 4-1 summarizes some important properties of diamond. The table was compiled from Yoder [Yode93], Spear and Dismukes [Spea94] and Field [Fiel92].

Table 4-1. Properties of diamond.

Property	**Value**	**Units**
Hardness	1.0×10^4	kg/mm^2
Strength, tensile	>1.2	GPa
Strength, compressive	>110	GPa
Coefficient of friction (Dynamic)	0.03	Dimensionless
Sound velocity	1.8×10^4	m/s
Density	3.52	g/cm^3
Young's modulus	1.22	GPa
Poisson's ratio	0.2	Dimensionless
Thermal expansion coefficient	1.1×10^{-6}	K^{-1}
Thermal conductivity	20	W/cm-K
Thermal shock parameter	3.0×10^8	W/m
Specific heat	0.853	J/gm-K

4.2 Pre-Coating Surface Treatment

The laser-based surface treatment under development at the University of Florida has not been previously introduced in the published literature. The process involves using a laser to modify a metallic substrate surface. This surface modification produces a uniform roughness which provides nucleation sites for the diamond coating growth and serves to increase the surface area of contact between coating and substrate. The increase in contact surface area improves the adhesion of the diamond layer and allows the interface to grade the surface stresses, effectively reducing the chances of premature debonding.

The surface treatment shows great promise for promoting diamond film growth on steel and other metallic surfaces. The dramatic thermal expansion mismatch between materials such as steel and diamond makes this endeavor extremely difficult. Silicon has been used successfully as a substrate for this process and results from this work are guid-

ing efforts on other metallic substrates. As an intermediate step, low-α metals[1]—such as molybdenum—are being tested with the process.

Figure 4-1 shows conceptually how the surface modification appears in cross section. Note how the diamond "seeds" sit inside the roughness "valleys." The seeds act as nucle-

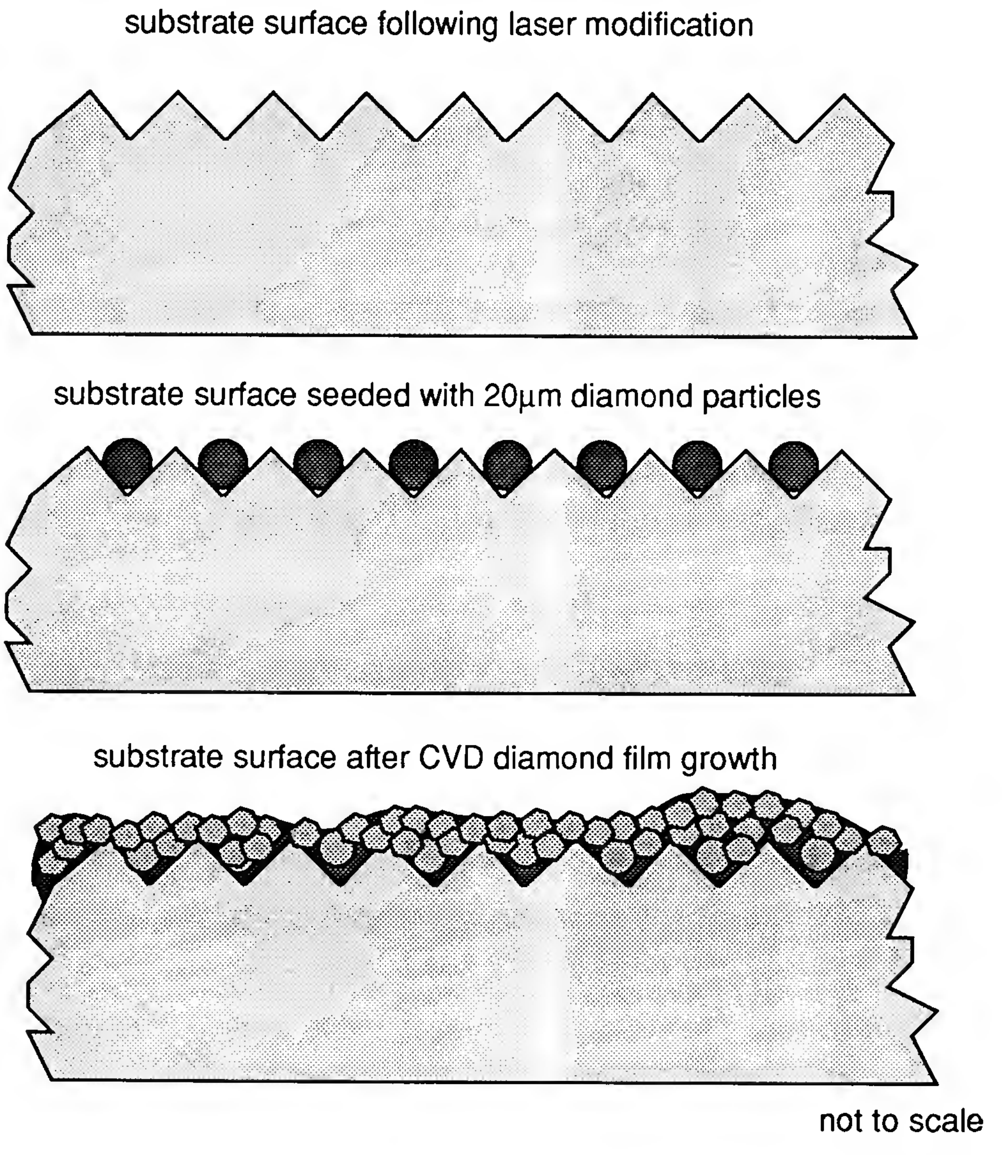


Figure 4-1. Cross sectional view of substrate surface following laser modification, seeding and film growth.

1. low thermal expansion coefficient

ation sites for the formation of the diamond film. This feature allows film growth to occur at a lower surface temperature than would otherwise be possible.

An electron microscope photograph illustrating a seeded substrate surface after a short growth duration is given in Figure 4-2. The surface roughness produced by the surface

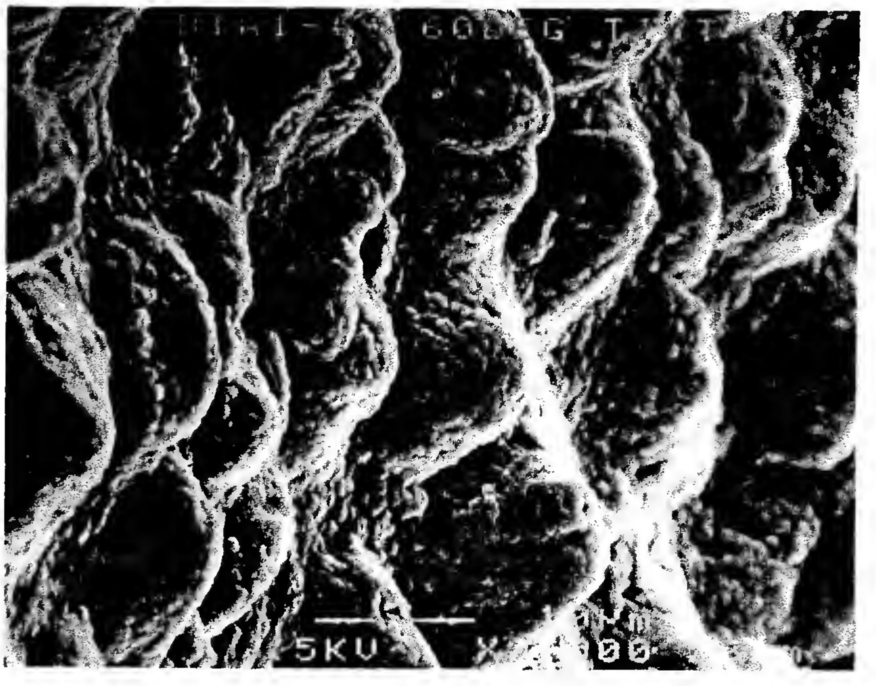


Figure 4-2. Seeded silicon substrate surface following a short duration (10 minute) diamond growth period.

preparation process is evident from the photo, but the diamond seeds make the surface look rougher than it is. The final surface topology after a complete diamond growth cycle is shown in Figure 4-3. The uniformity of the surface roughness is much more evident in this picture.

4.3 Chemical Vapor Deposition of Diamond

Diamond synthesis techniques have been available since the late 1950s (see Busch and Dismukes [Busc94]). The commercialization of synthetic high-pressure, high-temperature (HPHT) diamond grit occurred in 1959. This grit has had tremendous use in industrial polishing, cutting and grinding applications. The HPHT synthesis method essentially mimics

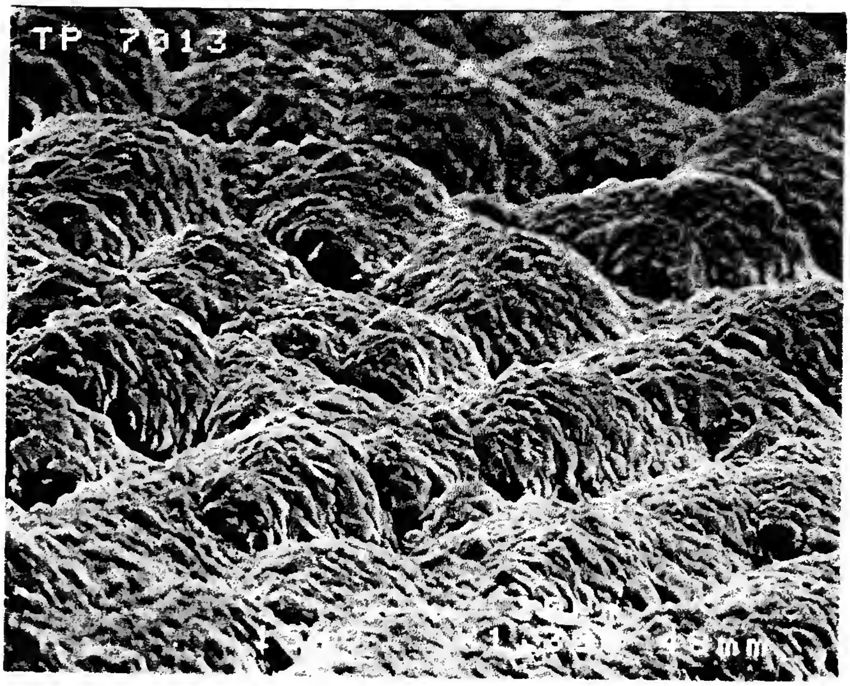


Figure 4-3. Final silicon substrate's surface topology following the complete diamond film growth cycle.

nature's way of producing diamond—only at much poorer quality. HPHT methods are basically only capable of producing grit.

The chemical vapor deposition (CVD) process, a low-pressure synthesis method, was successfully used to precipitate diamond-on-diamond seed crystals using carbon monoxide gas as a source of carbon, in 1952. This method actually predates the HPHT process by several years, but presented more challenges for commercialization.

CVD diamond growth methods use simpler apparatus less subject to mechanical wear and promises the production of physical forms of diamond other than powder (HPHT) (see Moustakas [Mous94]). One of the early drawbacks to CVD methods was the formation of graphite during diamond nucleation. Many variations have been tried in cleaning the graphite structures during diamond growth. Introducing hydrogen to the environment has been effective in "scrubbing" the diamond structures clean from graphite.

Researchers found that heating the substrate surface using a plasma source increased the diamond growth rate. This method helps decompose the methane gas into carbon

(methane has a high activation energy that caused slow growth rates). An illustration of a microwave-plasma CVD apparatus is shown in Figure 4-4.

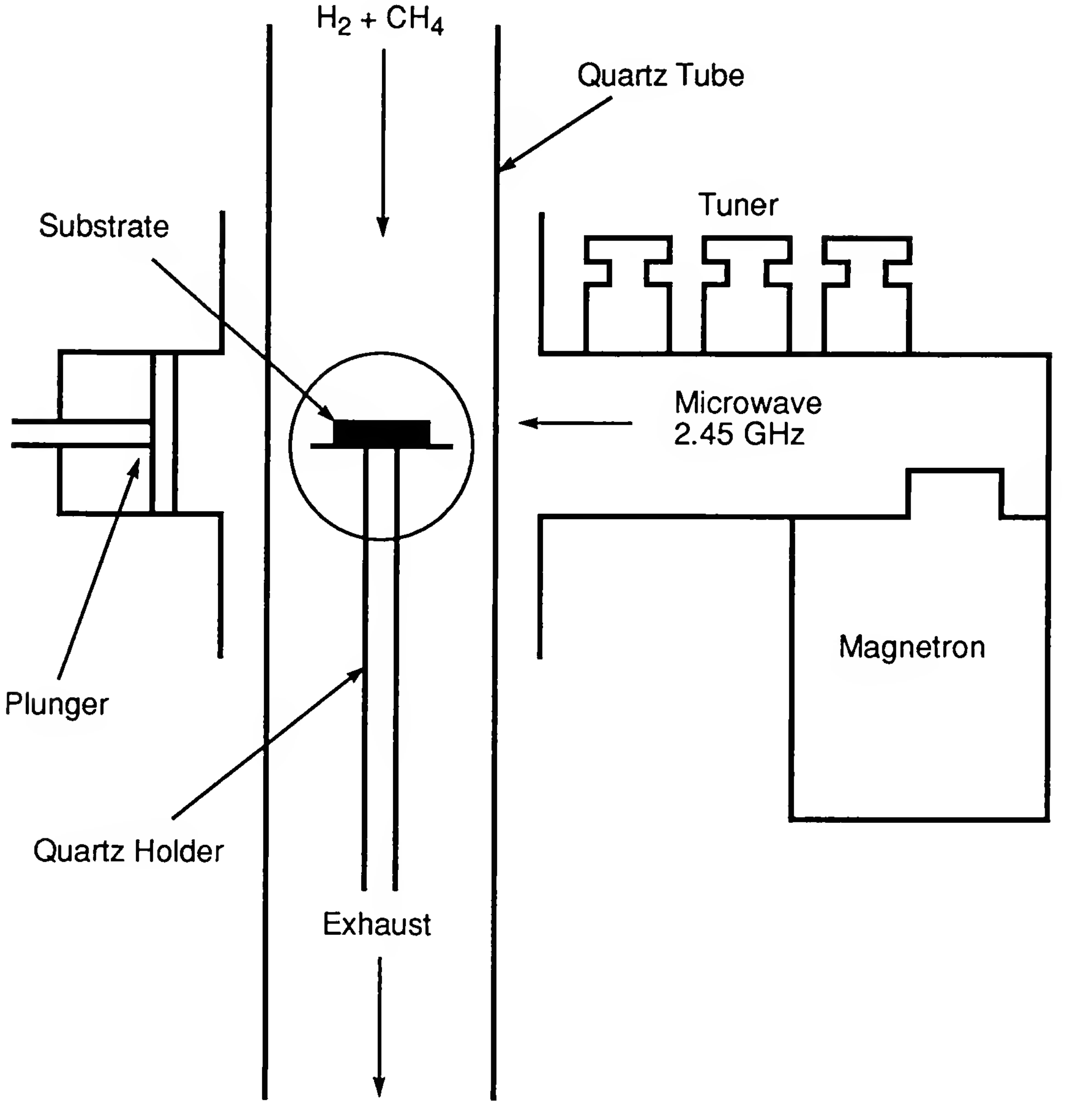


Figure 4-4. Schematic illustration of microwave-plasma CVD apparatus used for the growth of diamond films ([Mous94] p. 164).

Moustakas reports [Mous94] that microwave-assisted CVD methods are the most prevalent for diamond film growth. The process avoids contamination of film during growth and produces a higher plasma density over RF (radio frequency) methods. This results in higher concentrations of atomic hydrogen and hydrocarbon radicals necessary

for film growth. Since this method concentrates plasma at the center of deposition chamber, it prevents deposition of carbon on the quartz tube walls. Typical parameters for deposition using the microwave-plasma CVD, as reported by Moustakas ([Mous94] p. 163), are shown in Table 4-2.

Table 4-2. Deposition parameter space used in the growth of diamond films by microwave-plasma-assisted CVD.

Gas Mixture	Total Pressure (torr)	Microwave Power (W)	Substrate Temperature (°C)
CH_4 (0.5–2%)/H_2	5–100	100–700	700–1000

HPHT diamond is limited in application to planar surfaces. In this respect, HPHT diamond is no better than the natural diamond grit that it replaces. CVD diamond, however, ushers diamond applications to an exciting new level. "CVD diamond offers the potential to deposit large-area, conformable coatings with properties akin to that of natural diamond ([Busc94] p.592)." For the first time, diamond can be used as an engineered material—synthesized to meet specific topological and performance characteristics.

4.4 Cost Modeling

Cost modeling was used in this study to predict costs for applying the diamond coating. Busch [Busc94] used a cost modeling technique to assess factors influencing manufacturing costs in plastics fabrication processes. Busch and Dismukes applied this method to determine present and future costs in diamond coating [Busc94]. The cost modeling presented here is adapted from Busch and Dismukes' comparative assessment of CVD diamond manufacturing technology and economics. In their study, several CVD approaches are compared from near-term (3 to 5 years) and long-term (5 to 10 years) perspectives. Mature HPHT technology is used for a baseline. The technique used by Busch and Dismukes in their economic analysis is known as technical cost modeling (TCM). The sequential stages in TCM are illustrated in Figure 4-5.

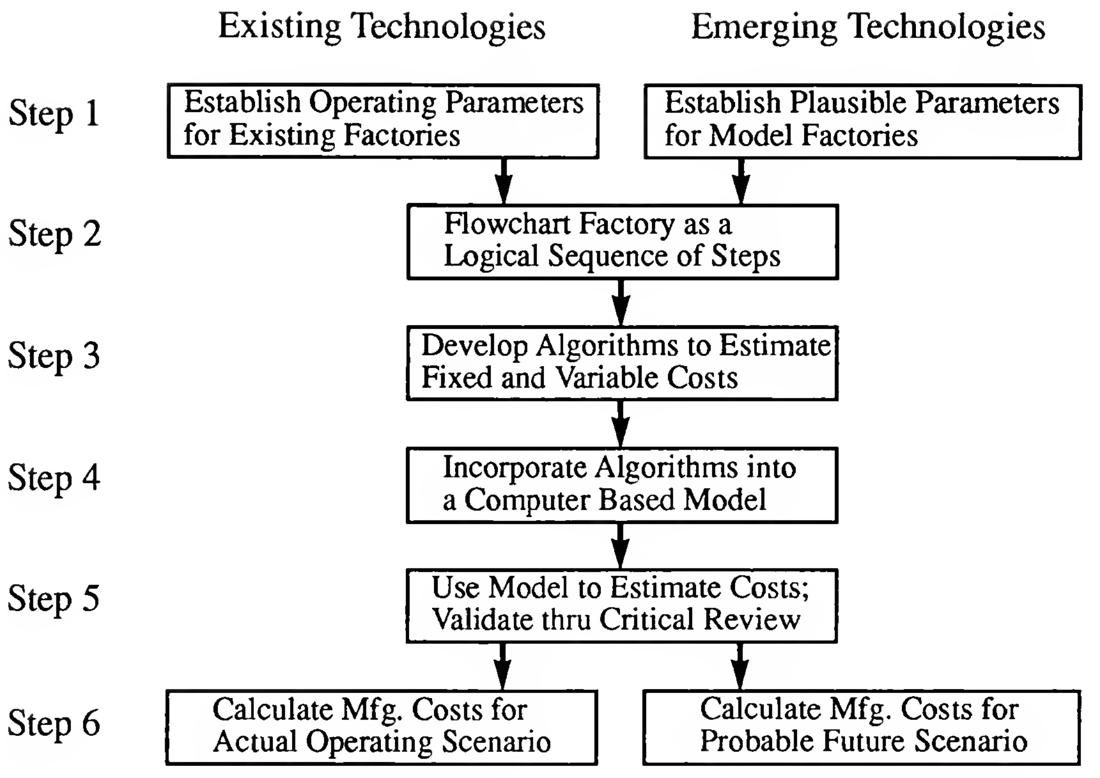


Figure 4-5. Sequential stages in technical cost modeling (TCM) ([Busc94] p. 586).

4.4.1 Near-Term Inputs

Significant near-term input parameters for the TCM are summarized in Table 4-3. The deposition parameters are expected or most probable values, as defined by Busch and Dismukes [Busc94]. The assumptions apply for deposition of 81 cm^2 diamond wafers, each 250 μm thick.

4.4.2 Long-Term Inputs

Table 4-4 assumes that significant technological improvements—feasible within 5 to 10 years—have occurred. The deposition capability includes 730 cm^2 diamond wafers measuring 250 μm thick.

Table 4-3. Near-term inputs ([Busc94] p. 593).

Input[a]	**Microwave**	**DC Arc Jet**
Hydrogen (%)	88.7	99.0
Carbon (%)	10.0	1.0
Oxygen (%)	1.3	NA
Hydrogen recycle rate (%)	0.0	0.0
Substrate area (cm^2)	81	81
Substrate cost	$0.01	$0.01
Coating thickness (μm)	250	250
Deposition rate (μm)	4	25
Direct laborers/station	0.2	0.2
Load/unload laborers/station	1	1
Load/unload time (min)	60	60
Total gas flow rate (sccm)	484	30250
Carbon capture factor (%)	8.0	8.0
Mass deposition rate (g/h)	0.11	0.71
Machine cost ($/station)	$350000	$350000
Machine power (kw)	15	90
Cooling water (GPM)	3	10
Building space (ft^2/station)	1000	1000

a. Dollar values are from 1992

4.4.3 Other Cost Factors

Other cost factors necessary to complete the analysis are included in Table 4-5. These figures are applicable to both the short-term and long-term inputs to the TCM process. Busch and Dismukes obtained the information for Table 4-6 on specific input costs for process gases by consulting directly with gas vendors and reviewing trade literature.

4.4.4 Near-Term Outputs

Table 4-7 summarizes near-term outputs for microwave and DC arc jet CVD deposition techniques. The wafer size is 81 cm^2 with a 250 μm thick deposition. The TCM

Table 4-4. Long-term inputs ([Busc94] p. 594).

Input[a]	Microwave	DC Arc Jet
Hydrogen (%)	88.7	98.0
Carbon (%)	10.0	2.0
Oxygen (%)	1.3	NA
Hydrogen recycle rate (%)	0.0	0.0
Substrate area (cm^2)	730	730
Substrate cost	$0.01	$0.01
Coating thickness (μm)	250	250
Deposition rate (μm)	15	60
Direct laborers/station	0.1	0.1
Load/unload laborers/station	1	1
Load/unload time (min)	120	120
Total gas flow rate (sccm)	6544	130900
Carbon capture factor (%)	20.0	20.0
Mass deposition rate (g/h)	3.85	15.42
Machine cost ($/station)	$550000	$700000
Machine power (kw)	75	350
Cooling water (GPM)	25	60
Building space (ft^2/station)	2000	2000

a. Dollar values are from 1992

reports a total-per-carat cost of $61.61 for the microwave deposition and $22.33 for the DC arc jet deposition. Since there are 0.2 grams per carat, these figures translate to $308 and $112, respectively, on a per gram basis. A cost estimate of $200 per-gram-deposited, the approximate average of the microwave and DC arc jet costs, was used throughout this study.

4.4.5 Long-Term Outputs

Table 4-8 summarizes long-term outputs for microwave and DC arc jet CVD deposition techniques. The wafer size is 730 cm^2 with a 250 μm thick deposition. The TCM reports a total-per-carat cost of $3.75 ($18.75/gm) for the microwave deposition and $2.42 ($12.10/gm) for the DC arc jet deposition. A cost estimate of $15 per-gram-deposited, the

Table 4-5. External cost factors ([Busc94] p. 595).

Direct wages[a]	\$13.33/h
Indirect salary	\$50000/year
Indirect: direct labor ratio	0.33
Benefits on wage and salary	35.0%
Working days per year	360
Working hours per day	24/day
Capital recovery rate	10%
Capital recovery period	5 years
Building recovery life	20 years
Working capital period	3 months
Price of electricity	\$0.100/kWh
Price of natural gas	\$6.50/MBTU
Price of building space[b]	\$100/ft^2
Price of cooling water	\$0.03/100 gal
Auxiliary equipment cost	15.0%
Equipment installation cost	35.0%
Maintenance cost	8.0%

a. Dollar values are from 1992

b. U.S. figure; cost does not include cost of land

Table 4-6. Process gas costs in the near and long term (1992 \$/m^3) ([Busc94] p. 595).

Gas	Near Term	Long Term
H_S	\$13.07	\$5.39
CH_4	\$19.50	\$6.57
C_2H_S	\$7.06	\$2.40
O_2	\$10.76	\$1.74

approximate average of the microwave and DC arc jet costs, would therefore be a reasonable long-term estimate for deposition costs—a reduction in costs by over a factor of 13.

Table 4-7. Near-term outputs ([Busc94] p. 595).

Output	**Microwave**	**DC Arc Jet**
Variable costs		
Material cost	$25	$238
Labor cost	$270	$60
Energy cost	$97	$92
Fixed costs		
Capital equipment cost	$945	$210
Building cost	$45	$10
Maintenance cost	$450	$100
Overhead labor cost	$30	$7
Cost of capital	$334	$79
Total cost ($/run)	$2196	$796
Total cost ($/cm^2)	$27.11	$9.83
Total cost ($/carat)	$61.61	$22.33

Table 4-8. Long-term outputs ([Busc94] p. 596).

Output	**Microwave**	**DC Arc Jet**
Variable costs		
Material cost	$39	$177
Labor cost	$73	$48
Energy cost	$133	$150
Fixed costs		
Capital equipment cost	$495	$210
Building cost	$30	$10
Maintenance cost	$246	$100
Overhead labor cost	$5	$2
Cost of capital	$184	$79
Total cost ($/run)	$1204	$776
Total cost ($/cm^2)	$1.65	$1.06
Total cost ($/carat)	$3.75	$2.42

CHAPTER 5

POTENTIAL APPLICATIONS

This chapter explores potential applications for the diamond coating process. Each application area discussion includes nomenclature, design objectives and design examples illustrating limitations with current materials. In each of these applications it will be demonstrated that thermal properties of the contacting materials are extremely important. Diamond's extremely high thermal conductivity will prove to be an excellent match for these applications.

5.1 Application to Gears

Successful gear system design requires balancing design parameters that affect tooth strength and surface durability. Surface failure results from interaction between wear mechanisms, contact fatigue and thermal effects. Many gears operate under conditions of boundary and mixed lubrication. In such cases tooth surface temperature rise is very important.

5.1.1 Gear Nomenclature

The following nomenclature is used in gear systems—corresponding equation numbers are in parenthesis:

ΔT_p = temperature rise on the pinion surface (5.12)
ΔT_g = temperature rise on the gear surface
$\dot{q}$ = heat flux per unit contact length (5.1)
f = coefficient of friction
W_N = normal force on the surface per unit length
T_q = input torque per inch of facewidth
F = facewidth
V_r = rolling velocity (5.2)

V_s = sliding velocity (5.3)
V_p = instantaneous tangential velocity at pinion contact point (5.4)
V_g = instantaneous tangential velocity at gear contact point (5.5)
ω_p = angular velocity of the pinion
M_G = gear ratio
ϕ = pressure angle
x = position of contact along the line of action measured from the pinion base circle
x_s = position x at the starting point of contact (5.11)
x_e = position x at the end point of contact
a = width of the contact band (5.6)
R_e = effective radius of the contacting cylinders (5.7)
K = thermal conductivity of the gear material
ρ = density of the gear material
c = specific heat of the gear material
ψ_p = heat partition coefficient for the pinion (assumed to be 0.5 for equal heat partition between the pinion and the gear)
ψ_g = heat partition coefficient for the gear (assumed to be 0.5 for equal heat partition between the pinion and the gear)
E_e = effective modulus of elasticity for the contacting materials (5.8)
d_p = pitch diameter of the pinion (5.9)
C_d = center distance
W_t = tangential load per inch of facewidth
N_p = number of teeth in the pinion.

5.1.2 Governing Equations for Gear Pairs

The following equations define relationships between the gear system parameters:

heat flux:

$$\dot{q} = \frac{\Delta T_p \sqrt{\pi K \rho c}}{2\psi_p \sqrt{\frac{a}{V_p}}} \tag{5.1}$$

velocity terms:

$$V_r = \frac{V_p + V_g}{2} \tag{5.2}$$

$$V_s = |V_p - V_g| \tag{5.3}$$

$$V_p = \omega_p C_d x \sin\phi \tag{5.4}$$

$$V_g = \omega_p \frac{C_d}{M_G} (1 - x) \sin\phi \tag{5.5}$$

contact zone width:

$$a = 2.15 \sqrt{\frac{W_t \cos\phi}{E_e} R_e} \tag{5.6}$$

effective radius:

$$R_e = C_d (1 - x) x \sin\phi \tag{5.7}$$

effective modulus of elasticity:

$$E_e = \frac{1}{\frac{1}{E_1} + \frac{1}{E_2}} \tag{5.8}$$

pinion pitch diameter:

$$d_p = 2\frac{C_d}{1 + M_G}. \tag{5.9}$$

5.2 Minimizing Contact Temperature in Gear System Design

Lin and Seireg [Lin85] proposed an optimum gear system design strategy that minimizes contact temperature while maximizing the load-carrying capacity of the system. The researchers used the following dimensionless relation for the temperature rise on the pinion surface:

$$\Delta T_p^* = \frac{\Delta T_p}{f\left(\frac{E_e \omega^2 W_t^3 C_d}{\rho^2 K^2 c^2}\right)^{1/4}}$$

$$= 0.77\left|\frac{(1+M_G)x-1}{M_G}\right|\left[\frac{\sin\phi}{(1-x)x^3\cos^3\phi}\right]^{\frac{1}{4}}\Psi_p. \tag{5.10}$$

Similar relations were developed for the dimensionless temperature rise in the gear, ΔT_g^*. The relationships for ΔT_p^* and ΔT_g^* are functions of the pressure angle ϕ, gear ratio M_G and the contact position x only. Cheng, et al. [Pati77,Wang81], found the minimum values of the surface temperature occur at the pitch point and the maximum values occur at the lowest point of contact on the dedendum. Lin and Seireg confirmed this result and further found that for "standard addendum, the maximum temperature rise calculated by the simplified formula for mixed lubrication occurs at the pinion addendum (starting point of contact) for gear ratios greater than one and at the gear addendum (end point of contact) for gear ratio less than one," ([Lin85], p. 554). For a gear set with standard addendum, the start point can be expressed as:

$$x_s = 1 - \frac{\sqrt{\left(M_G + \frac{2}{N_p}\right)^2 - M_G^2\cos^2\phi}}{(1+M_G)\sin\phi}. \tag{5.11}$$

The temperature rise at the starting point of contact, ΔT_p^* and ΔT_g^* , is illustrated in Figure 5-1. The graph indicates that for small number of teeth in the pinion, N_p, temperature rise on the surface can be quite high. The intermittent meshing of the teeth causes a cycle of high transient surface temperature rise followed by rapid cooling. Dooner and Seireg have shown this thermal shock cycle causes thermal stresses high enough to be become significant relative to contact stresses for the case of mixed lubrication. Figures 5-2 and 5-3 compare the nominal thermal and contact stress for standard gear teeth.

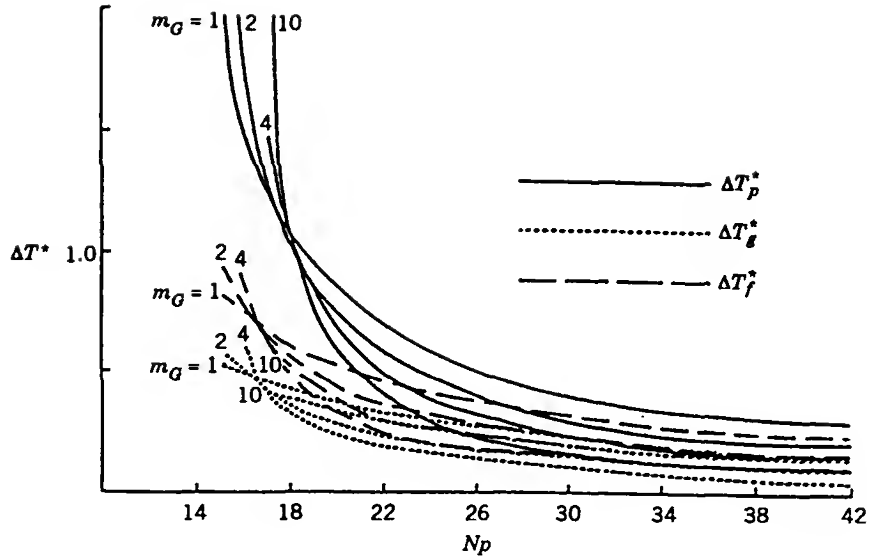


Figure 5-1. Temperature rise at the starting point of contact (ϕ = 20°) ([Doon95] p. 419).

5.3 Design of a Single Reduction Gear Set

Provided that center distance, C_d, is not pre-selected, a typical design objective would be that the gear set provides the required speed ratio and carries the given design load with the minimum volume of gear materials. Lin and Seireg state the problem as follows:

$$
\begin{aligned}
\text{given parameter} &= M_G,\ \omega_p,\ T_q,\ \phi, \text{ with standard addendum} \\
\text{decision parameters} &= N_p,\ C_d \\
\text{constraints} &= \Delta T_p \le \Delta T_g = 100\ °\text{F} \\
&\quad\ \sigma_{bmax} \le \sigma_{ba} = 50{,}000 \text{ psi} \\
&\quad\ \sigma_{max} \le \sigma_{Ha} = 150{,}000 \text{ psi} \\
&\quad\ x_s > 0 \\
\text{design region is} &= 18 \le N_p \le 40 \\
\text{bounded by} &\quad\ 5 \le C_d \le 45
\end{aligned}
$$

$$
\text{objective function} = \text{minimize } V = \frac{\pi}{4}\frac{1 + M_G^2}{(1 + M_G)^2}C_d^2.
$$

The equations governing the constraints are defined as follows:

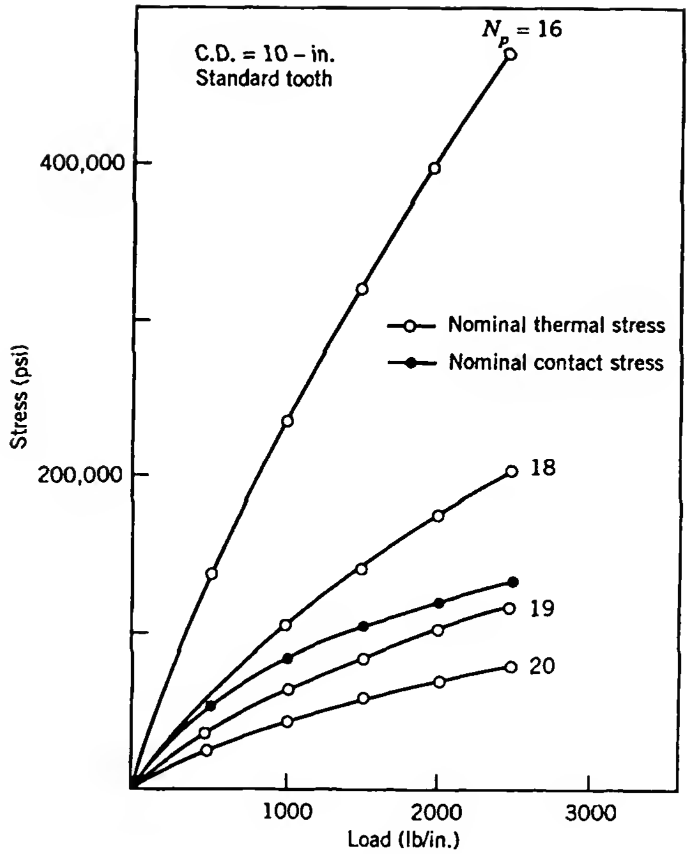


Figure 5-2. Nominal thermal and contact stress for standard gear teeth ([Doon95] p. 420).

$$\Delta T_p = 0.77f\left(\frac{E_e\omega^2 W_t^3 C_d}{\rho^2 K^2 c^2}\right)^{1/4}\left|\frac{(1+M_G)x_s - 1}{M_G}\right|\left[\frac{\sin\phi}{(1-x_s)x_s^3\cos^3\phi}\right]^{\frac{1}{4}}\psi_p \tag{5.12}$$

2. maximum bending stress:

$$\sigma_{b_{max}} = 13.5\frac{W_t N_p}{\pi^2 C_d}(1+M_G) \tag{5.13}$$

3. maximum contact stress:

$$\sigma_{H_{max}} = 0.417\sqrt{\frac{W_t E_e}{R_e \cos\phi}} \tag{5.14}$$

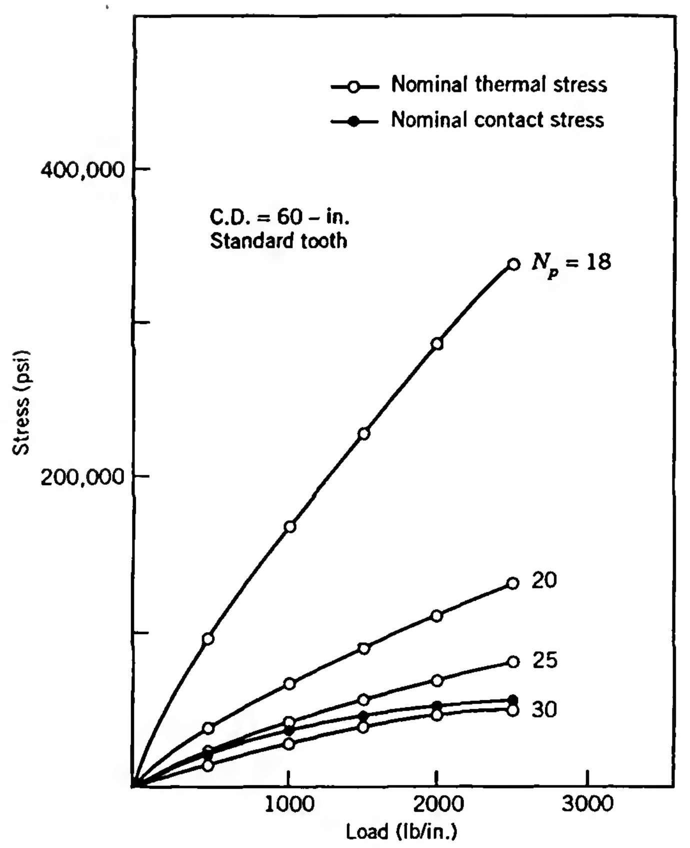


Figure 5-3. Nominal thermal and contact stress for standard gear teeth ([Doon95] p. 421).

4. tangential load per unit length:

$$W_t = \frac{T_q(1+M_G)}{C_d}. \tag{5.15}$$

5.3.1 Example Single Reduction Gear Set

The following numerical example, using a steel gear, is taken from Lin and Seireg:

M_G = 2
T_q = 30,000 in. lb/in.
ω_p = 1750 rpm
f = 0.045
ψ_p = 0.5
ϕ = 20°
K = 26 Btu/hr-ft.-°F
ρ = 490 lbm/ft.^{3}
c = 0.10 Btu/lbm-°F
E_e = 15,000,000 psi.

The design constraints are the same as defined in section 5.3.

An optimization strategy using the steepest gradient and golden-section methods was used to obtain the following solution:

N_p = 40
C_d = 36.65 in.

The constraint values are as follows:

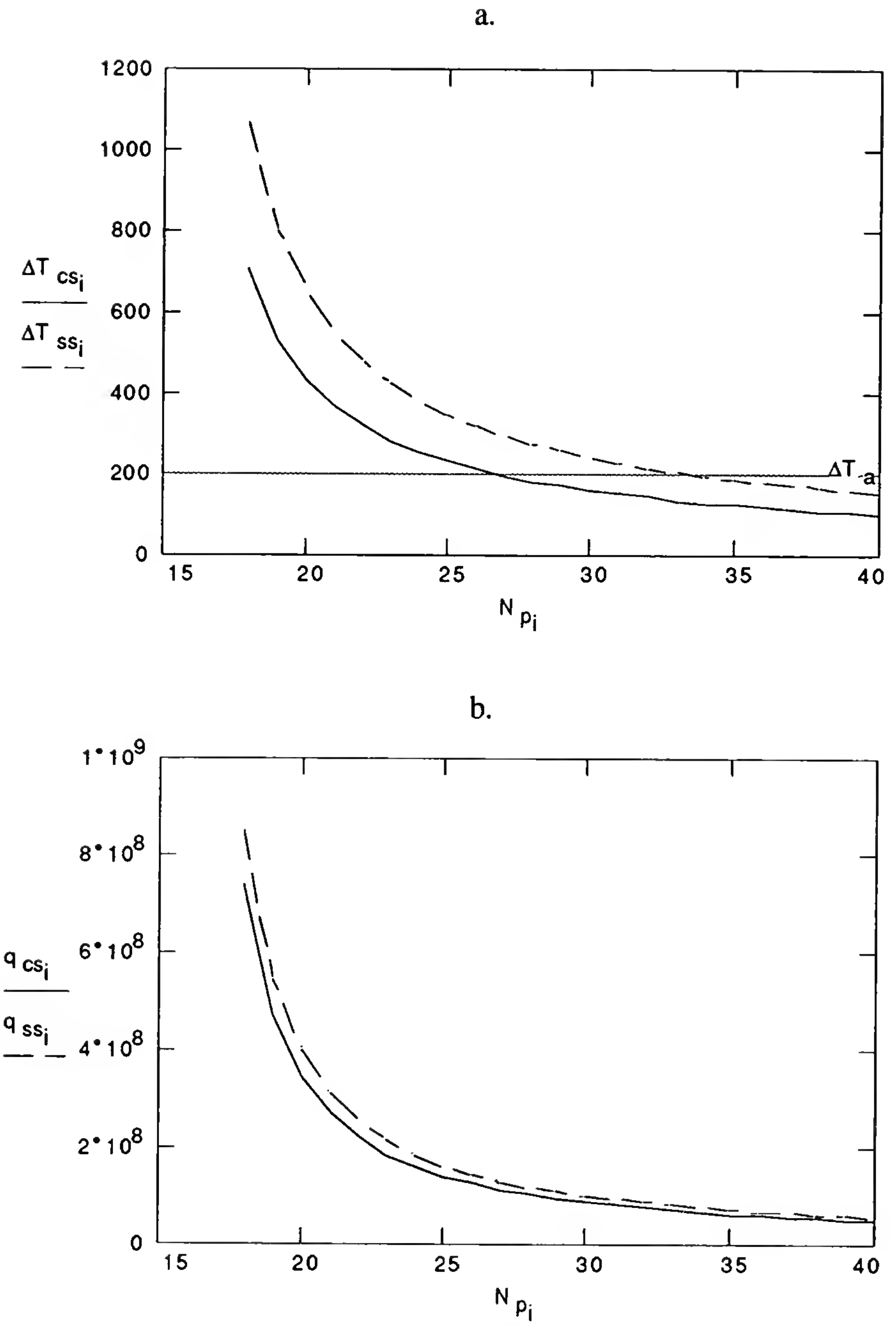
ΔT_p = 200.0 °F
σ_{bmax} = 11.0 kpsi
σ_{max} = 58.1 kpsi
x = 0.202.

5.3.2 Comparison of Carbon Steel and Stainless Steel Gears

Many corrosive environments preclude the use of high-strength carbon steel gears. In these environments it would be beneficial to use stainless steel gears. Unfortunately, stainless steels have relatively low thermal conductivity compared to carbon steels. A typical value for stainless steel thermal conductivity is $16.6 \frac{\text{watt}}{\text{m-°C}}$ whereas a typical carbon steel

used in gear applications has a value of $46.7 \frac{\text{watt}}{\text{m-°C}}$. Temperature rise on the gear contact surfaces is dependent on the magnitude of $K\rho\, c$ as shown in Equation (5.12).

The difference in temperature rise on the pinion surface is compared for carbon and stainless steel gears under the same operating conditions for N_p ranging from 18 to 40 teeth in Figure 5-4. The allowable temperature rise for this example is 200 °F. Note that while the heat flux generated in the contact zone for each gear are very close, the temperature rise values differ by approximately 50 °F for the best case ($N_p = 40$). This temperature difference limits the choices of the designer when selecting components for this application. To maintain the same gear ratio, the center distance between the pinion and the gear must be adjusted. If center distance is a design constraint, then it may not be possible to use a stainless steel gear pair. The most important constraint, however, on the use of stainless steel in sliding contacts is the tendency for galling. Consequently, the use of an appropriate coating layer is mandatory in this case.



carbon steel properties: $K\rho c$ = 1570 lbf.^{2}/°F-in.^{2}-sec., E=30x10^6 psi
stainless steel properties: $K\rho c$ = 680 lbf.^{2}/°F-in.^{2}-sec., E = 30x10^6 psi
C_d = 10", W_t = 2500 lb./in., N = 1800 rpm, M_G = 5, and ϕ = 20°

Figure 5-4. Comparison of carbon steel and stainless steel gears under the same operating conditions.
a) Temperature rise (°F) in the pinion; b) Heat flux (watt/m^2).

CHAPTER 6
THE MATCHING PROCESS

6.1 The Matching Process Role in the Technology Utilization Methodology

This study determined that finding suitable applications for new laboratory discoveries is a two-level process. First, qualitative requirements of potential applications are compared with qualitative capabilities of the new discovery. Once a qualitative attribute match is established, a quantitative design analysis with cost/benefit evaluation can be performed. Multiple application matches can be ranked based on the results of cost/benefit analyses. The ranking suggests where the best return on investment can be realized for further development of the new technology. The framework for the process is illustrated in Figure 6-1.

In this chapter, a hierarchical structure is proposed for capturing qualitative and quantitative attributes for technologies. These technologies represent either new discoveries or potential applications. Next, a knowledge base for managing the attribute structure is presented. The knowledge base, developed in the CLIPS expert system shell, includes capabilities for pattern matching on attributes within a rule-based framework.

A detailed procedure for building the potential applications and emerging technologies knowledge bases is presented in Chapter 7. A sample run through the qualitative matching procedure is also included.

6.2 The Structure of the Developed Matching Process

The matching process discussed in this chapter is intended to present the basic approach for qualitative matching. The implementation, however, is heavily biased to the demonstrated process. Consequently the matching rules are limited in scope since the

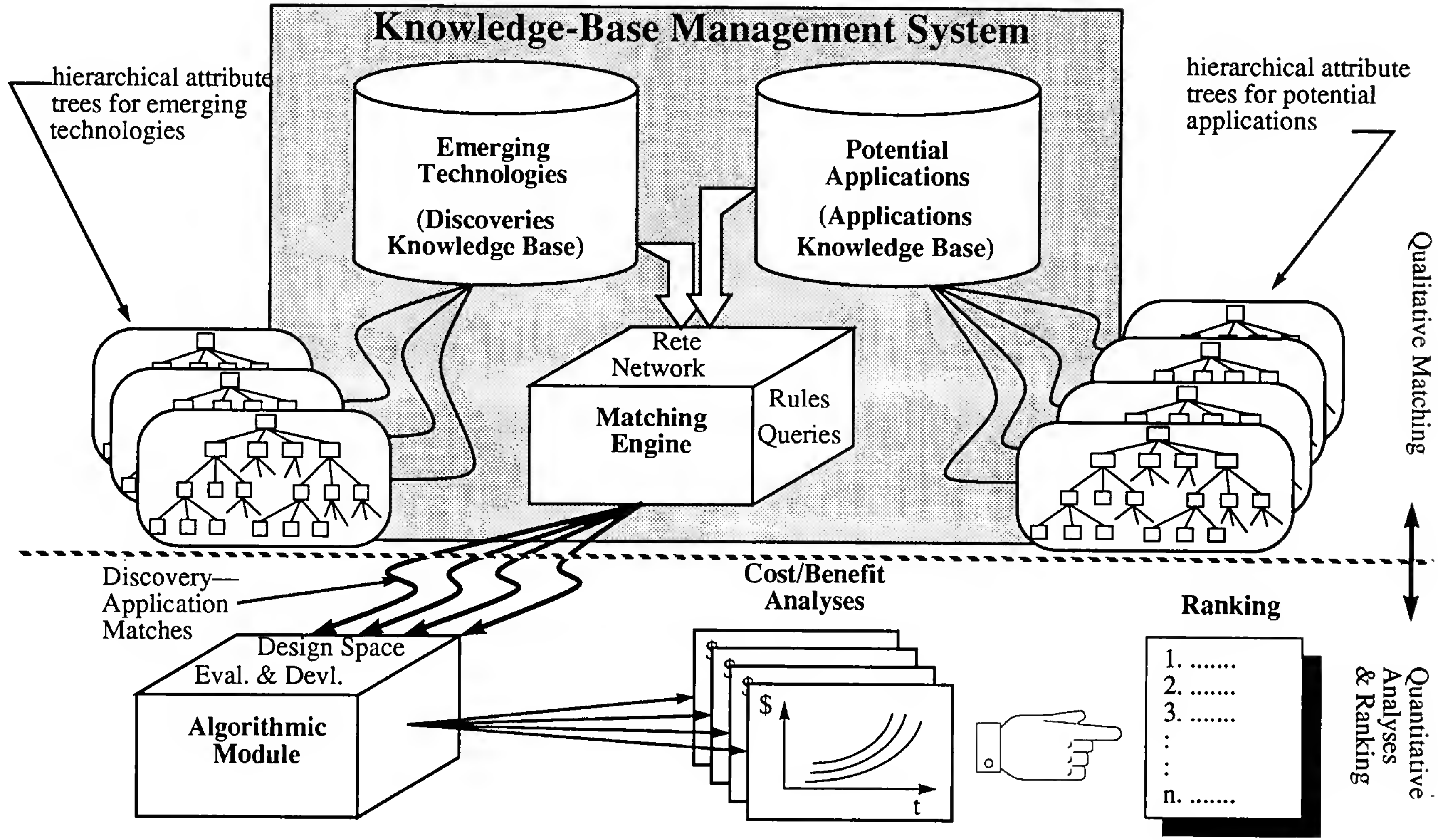


Figure 6-1. Framework of the algorithmic/heuristic technology evaluation process.

objective is to demonstrate the approach rather than the rigorous development of a general matching scheme.

6.2.1 Creation of a Structured Attribute Representation

Central to the development of the matching strategy was the creation of a structured representation for the attributes of emerging technologies—discoveries—and their potential applications. It was necessary to capture a variety of information in this attribute structure, including the following:

- material attributes
- manufacturing attributes
- equations governing the use of the technology
- qualitative requirements for the potential applications.

A common structure was developed to represent a technology—discovery or potential application. A technology's attributes were decomposed progressively from general categories such as material attributes down to atomic property values such as thermal expansion coefficient. An atomic property cannot be further broken down into finer specializations.

Qualitative properties defined for discoveries are considered to be capabilities, whereas qualitative properties defined for potential applications are considered to be needs or requirements. This distinction between discovery capabilities and application needs makes it possible to develop a straightforward strategy for matching applications to discoveries.

6.2.2 Matching Atomic Values

The developed matching strategy compares atomic values of application attributes with atomic values of discovery attributes. Each atomic match of the discovery-application pair enters a match table where the fit between the capability and the need is evaluated

qualitatively. The match is considered strong or weak depending on how closely the criteria correspond.

6.2.3 Rating Discovery-Application Matches

Once all the attributes are matched, it is possible to judge how well the individual applications fit the discovery. A simplified qualitative rating scheme based on the percentage of strong matches relative to total matches for a given discovery-application pair was implemented. This method worked acceptably for the considered case study, but requires modification before it can be generalized. It is feasible to develop an attribute weighting scheme such that capability-need matches can be emphasized or de-emphasized for different discovery-application pairs. This capability was not implemented in the prototype matching system.

6.2.4 Prototype System Development

The developed matching strategy was implemented in a knowledge base system to demonstrate the technique. The following steps were involved in developing the prototype system:

1. a structure for the discovery and application attributes was developed and encoded;
2. a rule-driven mechanism based on atomic discovery-application attribute matching was developed and encoded;
3. a match table structure was developed and implemented to store and evaluate discovery-application attribute matches;
4. an application framework was developed and implemented to manage the matching process;
5. the knowledge base was populated with discovery and potential application attributes.

Steps 1 and 2 are discussed in this chapter and the remaining steps are developed in the next chapter.

6.2.5 Road Map for the Remaining Matching Process Discussion

Development and implementation details regarding the technology structure are detailed in this chapter. The principles behind the rule-driven matching engine are reviewed and matching engine inference strategies are compared. An introduction to the knowledge base system used to develop the prototype matching application completes the chapter. Chapter 7 details the construction of the prototype qualitative matching application, the process of populating the knowledge base and discusses the application system's execution.

6.3 Elements of the Matching Process

6.3.1 Technology Attribute Hierarchy Development

The American Heritage Dictionary of the English Language defines technology as the following:

1. a. The application of science, especially to industrial or commercial objectives. b. The scientific method and material used to achieve a commercial or industrial objective.
2. The body of knowledge available to a civilization that is of use in fashioning implements, practicing manual arts and skills, and extracting or collecting materials (*Anthropology*).

Definition 1 applies for the purposes of this study. The challenge here is to decompose a technology into attributes that can be compared with other technologies to find attribute matches. This study deals with two types of technology which are broadly labeled as DISCOVERY and APPLICATION. A DISCOVERY technology type represents a new laboratory discovery. An APPLICATION technology type represents a potential application for a DISCOVERY. It is useful to have a universal technology attribute hierarchy that can capture the attributes for both DISCOVERY and APPLICATION technology types.

6.3.1.1 Technology attribute categories

The proposed methodology groups technology attributes into five broad categories. This study looks at technology transfer from the perspective of turning new discoveries

into products. Therefore, the technology attribute groups proposed have a physical product objective. This physical product objective is further limited to engineered products. This class of products is aimed at meeting societal and customer needs and thus precludes the application to pure aesthetic products. With this qualification or product focus, the technology attribute categories are material attributes, manufacturing attributes, governing equations, qualitative requirements and other attributes. These attribute categories are considered to be second-level attributes. Although the tabulated attributes are specific to the considered problem, the approach is applicable to any technology under study.

Material attributes. Material attributes are composed of physical properties, economic properties and other properties. Physical properties deal with how materials in a technology interact with the physical world. These interactions may be thermal, mechanical or environmental. Subclasses within the physical properties attribute class are defined in Table 6-1. Economic properties may include such areas as cost modeling for the materials involved. These models refer to raw material costs rather than processing costs. Processing costs more logically fall under the manufacturing attributes branch of the technology tree. The other properties branch includes miscellaneous material properties not associated with the previous two categories.

Manufacturing attributes. Manufacturing attributes include any production-related properties and property classes relevant to the technology. The process subclass is the only currently defined specialization of the manufacturing attributes class. The process class includes information such as the economic cost model, parameters of the process, process methods and process limitations. Subclasses within the attribute class for Process properties are defined in Table 6-2.

Governing equations. The governing equations subclass captures the empirical relationships, design parameters and constants used in describing the behavior of the technology. Empirical relationships might include predictive equations that relate qualities such as thermal stress to various attributes or parameters used in analyzing the technology's

Table 6-1. Subclasses defined within the Physical properties class for a sample knowledge base.

Class Subclass	Description
Thermal	class representing how a material's physical properties interact thermally with the environment
conductivity, K	heat transfer proportionality constant
specific heat, C	coefficient representing the energy storage capacity per unit density
expansion coefficient, α	coefficient representing the increase in linear material growth per change in material temperature
diffusivity, α_{diff}	measure of heat transport relative to energy storage
shock resistance	qualitative value of the thermal shock resistance
usage temperature, T_{max}	maximum useful material operating temperature range
Mechanical	class representing how a material's physical properties interact mechanically with the environment
Young's modulus, E	proportionality constant relating stress to strain; a measure of the stiffness or rigidity of a material
shear modulus, G	proportionality constant relating stress to strain; a measure of the stiffness or rigidity of a material
Poisson's ratio, ν	ratio of lateral to (-)axial strain
density, ρ	mass per unit volume
coefficient of friction (wet), μ_{lub}	kinetic coefficient of friction for the material sliding on itself with a lubricant film
coefficient of friction (dry), μ_{dry}	kinetic coefficient of friction for the material sliding on itself without a lubricant film
crack resistance	qualitative value of the material's crack resistance
roughness, R_a	peak-to-valley amplitude
hardness, H	Brinell hardness
toughness, K_c	fracture toughness or critical stress-intensity factor
Environmental	class representing how a material's physical properties interface with the environment
lub. film thkns., h_o	range of lubricating thicknesses
lub. compatibility	a list of compatible lubricants
oxidation resistance	qualitative measure of the material's oxidation resistance

Table 6-2. Subclasses defined within the Process properties class for a sample knowledge base.

Class Subclass	Description
Economics	class representing economics of the manufacturing process attributes
material	cost of materials per unit weight of finished product
labor	cost of labor per unit weight of finished product
power	cost of power per unit weight of finished product
overhead	cost of overhead per unit weight of finished product
capital equipment	cost of capital equipment per unit weight of finished product
Parameters	class representing manufacturing process parameters
pressure	process operational pressure
atmosphere	process atmosphere
surface temperature	surface temperature of substrate in process
Methods	class representing the manufacturing process methods
preparation	preparation/setup methods for process application
application	application methods used for process
finishing	finishing methods used for process
Limits	class representing manufacturing process limits
thickness	range of thicknesses that can be applied with the process
spatial requirements	required space to setup and operate processing equipment
deposition rate	thickness/area/volume of material that can be applied per unit time

design applications. Design parameters specify which variables used in an empirical relationship that are typically varied during the design process to achieve a suitable result.

Qualitative requirements. The qualitative requirements category only applies to APPLICATION technology types. This class describes qualitative information relative to a technology. These qualitative requirements may be further decomposed into specific attributes as needed for matching. For example, a gear application may require a high slide-to-roll contact ratio. This need may be decomposed into a series of specific require-

ments, such as a material with high thermal conductivity, low coefficient of friction and high hardness.

Other attributes. The other attributes category is defined for completeness. Some technology attributes may not be appropriate for the other second-level attributes. The other attributes category is a convenient location for such.

6.3.1.2 Skeleton for a technology hierarchy

A partial skeleton for a technology hierarchy, or technology tree, is illustrated in Figure 6-2. Note that not all leaves of the tree are populated. The Technology class is the root node of the tree and is said to be at level one. As each branch is explored level by level, the attributes become more specific. Thus the surface temperature leaf at level 5 is more specific than the process attribute at level 3.

6.3.2 Object-Oriented Implementation

The tree structure developed in the previous section suggests an object-oriented programming (OOP) language is appropriate for modeling and implementation. There are five primary characteristics that an OOP system must possess: abstraction, encapsulation, inheritance, polymorphism and dynamic binding [Booc91,Somm92,Giar91]. Booch defines OOP as follows:

> Object-oriented programming is a method of implementation in which programs are organized as cooperative collections of objects, each of which represents an instance of some class, and whose classes are all members of a hierarchy of classes united via inheritance relationships [Booc91] (p. 36).

6.3.2.1 OOP system characteristics

Abstraction. An abstraction is a high-level, intuitive means of representing a concept. We as humans use abstraction to grasp complex concepts. A well-developed abstraction

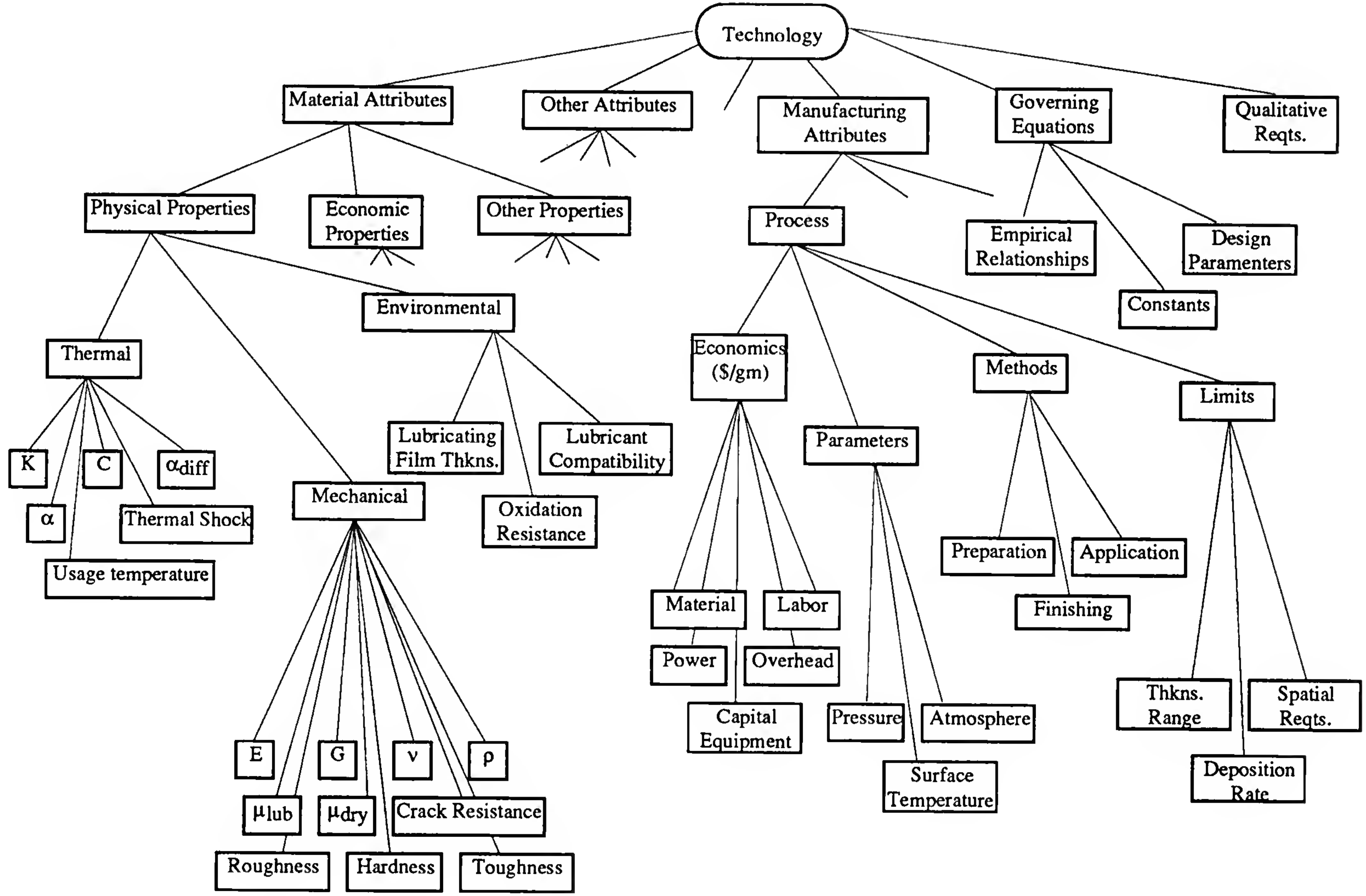


Figure 6-2. Structure for the knowledge base technology class hierarchy tree.

emphasizes details that are important to the user and suppresses those details that divert attention or confuse the subject. Booch defines abstraction as follows:

> An abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of objects and thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer [Booc91] (p. 39).

Encapsulation. Encapsulation, also known as information hiding, is the process of hiding the implementation details of an object using a well-defined interface. Whereas abstraction concentrates on how an object is viewed from the outside, encapsulation concentrates on the inner workings of the object. Booch summarizes encapsulation as follows:

> Encapsulation is the process of hiding all of the details of an object that do not contribute to its essential characteristics [Booc91] (p. 46).

The only access to an encapsulated object is through its message handling system. This means that it is only possible to "get" data from an object if that object has a *message-handler* (also known as a *method*) called "get." This access to objects only through their predefined message-handlers prevents unintended corruption of data and minimizes the possibility for unwanted side effects.

Inheritance. Inheritance is a mechanism to describe a subclass in terms of one or more superclasses. Single inheritance systems only permit a subclass (child class) to inherit characteristics from one parent class. Multiple inheritance systems permit a child class to inherit characteristics from one or more parent classes. Classes are arranged hierarchically such that the most general classes are at the top and the most specific at the bottom. Figure 6-2 illustrates a simple single inheritance hierarchy. Figure 6-3 represents a simple multiple inheritance hierarchy for quadrilaterals.

Inheritance also speeds up the development of software systems because it enables code to be easily reused. Many commercial and public domain object libraries are available to software developers.

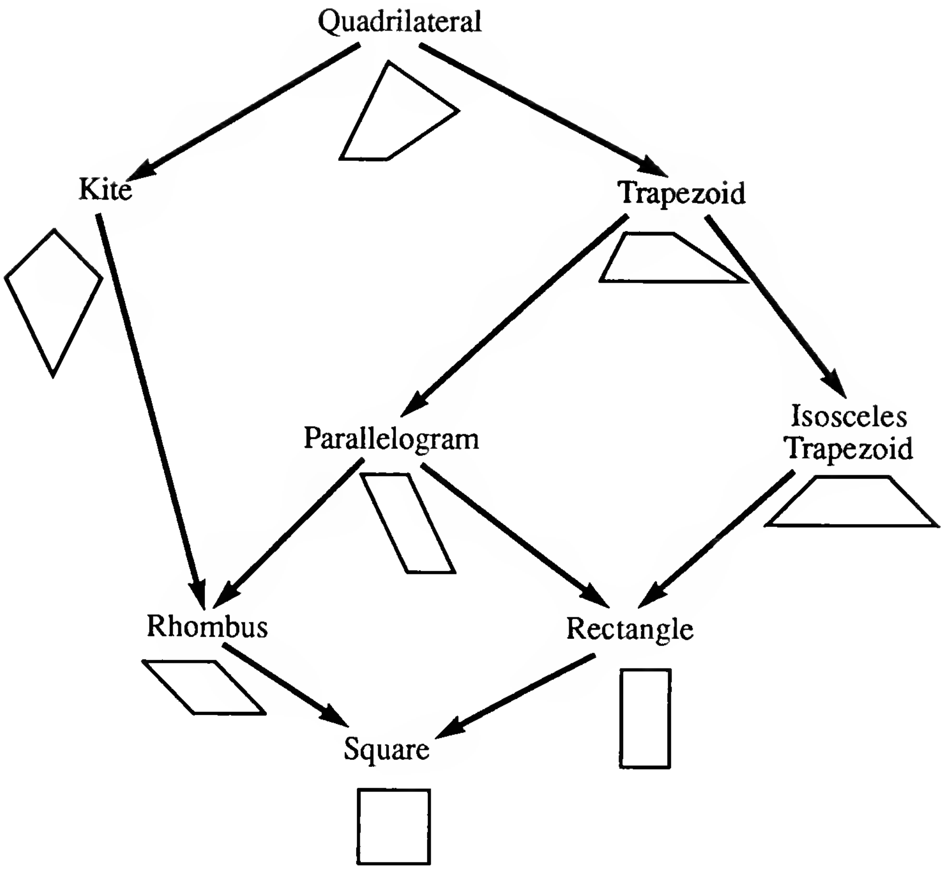


Figure 6-3. The quadrilateral hierarchy illustrating multiple inheritance ([Giar93] p. 101).

Polymorphism. Polymorphism is the capability for multiple objects to respond to the same message in different ways. For example, the "+" operator may be defined such that for a "+" message sent to two number objects, the sum of the numerical values is returned and that for a "+" message sent to two text string objects, the result returned is the concatenation of the strings into a single string. The process of assigning multiple behaviors to an operator, such as +, -, /, *, >, etc., is referred to as *operator overloading*. If a function name is used instead of an operator, then the *function* is said to be *overloaded.*

Dynamic binding. Dynamic binding defers selection of a reaction to a message until run-time. This means that the argument types passed to a function or method do not have to be known at compile time. Instead, the system will determine the appropriate behavior to a message when it is received. Dynamic binding works in conjunction with the poly-

morphism concept. In the case of the overloaded "+" function described previously, it would not be necessary to define the argument types passed to "+" at compile-time.

6.3.2.2 OOP advantages

Using an OOP language for developing large software systems has well-documented advantages [Booc91, Somm92]. A key advantage for the work presented in this study is the ability to manage a complex system through the use of data abstraction, also known as information hiding. The inheritance feature of OOP languages is invaluable for capturing the structure of the technology tree attribute relationships. Another important overall feature of OOP languages is the simplification of implementing object-oriented designs (OOD). OOD is a modern software engineering paradigm best used for developing large, complex software applications.

6.3.2.3 OOP terminology

The following terms are specific to OOP and are used throughout this chapter:

1. class: a template for describing the common attributes (slots) and behaviors of a group of objects called instances of a class.
2. object: an instance of a class.
3. instance: an object that can only be manipulated with messages.
4. message: the mechanism to manipulate an object.

The technology tree can be modeled as a series of parent–child relationships, where the parent represents the superclass and the child represents a subclass of a parent. Referring back to Figure 6-2, Material Attributes is the parent class of Physical Properties, Economic Properties and Other Properties.

6.3.2.4 Class structure of the technology tree

Class definition. A class is defined by its data attributes, or slots, and its message-handlers. Objects pass control from one to the next via messages. Message-handlers are an object's interface to other objects. This structure prevents objects without a specific func-

tion from corrupting the data from another. An example of a message sent to an object might be a print request. A typical class definition is shown in Figure 6-4.

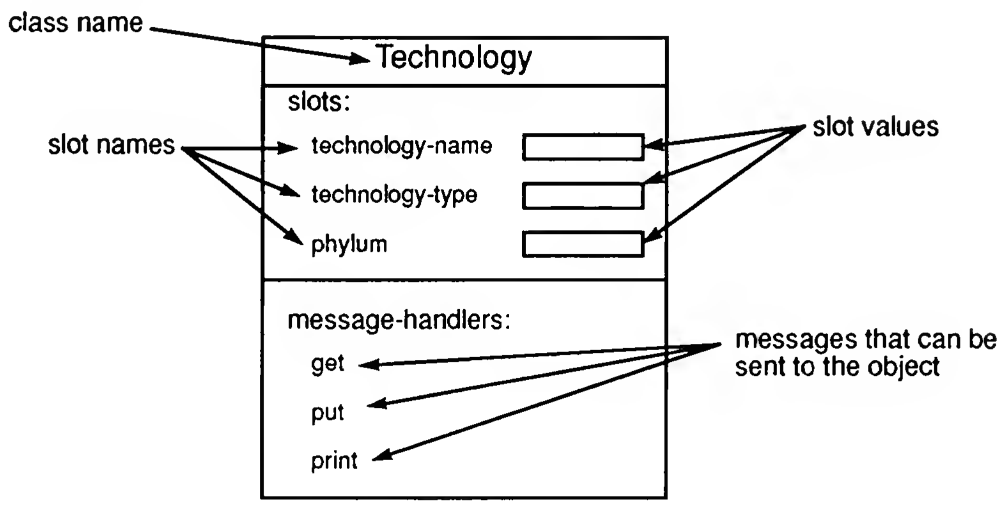


Figure 6-4. Structure of a class definition.

Inheritance mechanism. A child class is defined in terms of one or more parent classes, which may have in turn inherited properties from their parent classes, etc. Figure 6-5 demonstrates a typical inheritance mechanism. In this case the Material-Attribute class inherits the slots "technology-name", "technology-type" and "phylum" from the Technology superclass and the slot "classification" from the Level-1-Attribute class. Note that the classes are shown connected with "is-a" links. This is read as the Material-Attribute is-a Technology and the Material-Attribute is-a Level-1-Attribute.

6.4 The Matching Engine

The new technology-application matching engine was developed using an expert system. There are three components to the expert system (also referred to as the knowledge-based system) used in this study. First is the knowledge base, which is composed of rules (heuristics). Next is the data, which is comprised of facts and instances. And lastly, the inference engine, which is provided by the expert system shell to apply the knowledge to the data.

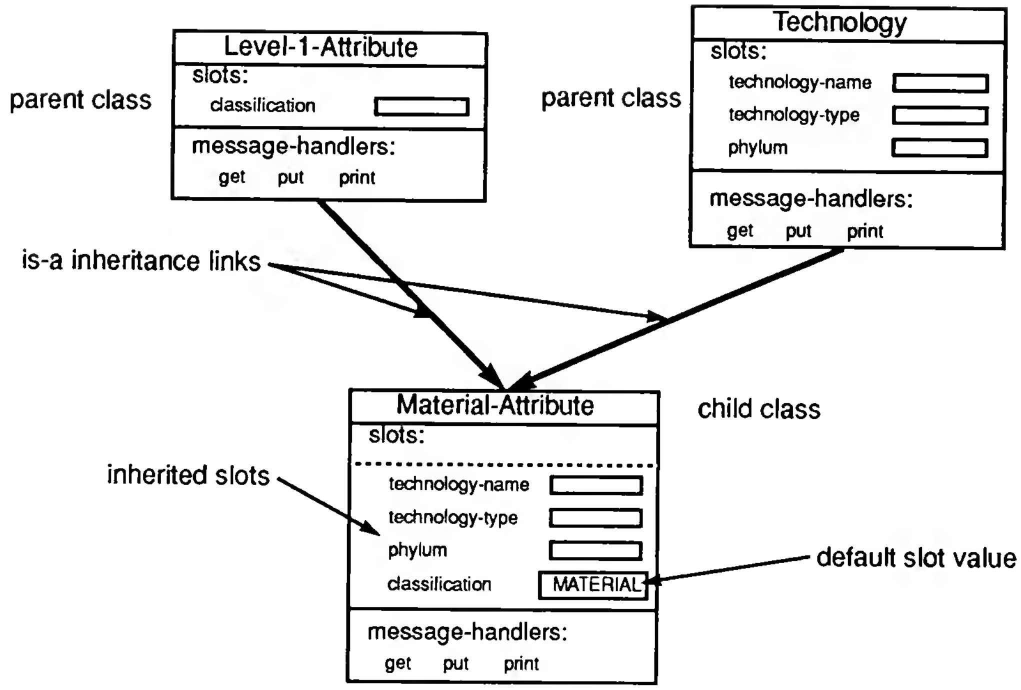


Figure 6-5. Inheritance mechanism for a child class with multiple parent classes.

This section develops details describing the qualitative matching process. First, an overview of the process is presented. Next, the expert system shell used to implement the matching engine is described. And lastly, sample qualitative rules used in the process are given.

6.4.1 Matching Process Overview

Information regarding new technologies and potential applications is encoded within the knowledge-base management system in the form of explicit facts, instances of the technology class hierarchy and rules which manipulate the facts and instances. The matching process searches for fact and instance patterns that match rule conditions. If a rule matches facts and/or instances, then the rule takes some consequent action that may include retracting certain facts, asserting new ones, manipulating instances or launching a procedural algorithm.

When many rules interact with many facts and instances, considerable computation time may be consumed. A strategy must be employed to determine which rules fire first and also minimize computational effort each time the state of the knowledge base changes. One way to decrease the computational effort is to implement an efficient pattern matching network. The network used in this study is the Rete, which was developed by Forgy [Forg82] for the OPS5 production system.

6.4.1.1 Facts

Facts are used in an expert system as the data about which rules reason and represent the current state of the world. The world in this case exists within the confines of the knowledge-base management system. Facts are transient data within an expert system and may therefore be asserted, retracted (deleted) or modified. Facts may take the following form:

```
(light red)
(signal walk)
(car stop).
```

The facts listed here might be part of an expert system that directs traffic. When these facts are manipulated via rules within an inference engine, then the transient nature of facts within the expert system will be more clearly shown.

6.4.1.2 Instances

Classes only define structure and cannot contain data. Therefore, classes cannot be pattern matched or directly manipulated by the inference engine. Instances, which are objects based on user-defined classes, are used to create copies of classes that can be manipulated. Instances may be pattern matched and can be modified via messages. For example, a print message might be sent to an instance, resulting in a list of the object's contents output to the system interface. Concrete examples of instances are developed later in this chapter.

6.4.1.3 Rules

Rules are collections of conditions and actions. When patterns within the rule conditions are matched, actions are taken. Rules have a format based on an if-then construct. A simple rule is based on the following pseudocode:

```
(Rule <rulename>
   (<pattern 1>)
   (<pattern 2>)
      :
      :
   (<pattern n>)
=>
   (<action 1>)
   (<action 2>)
      :
      :
   (<action m>)
); end of rule.
```

This rule can be read as "if <pattern 1> and <pattern 2> and . . . <pattern n> then do <action 1>, <action 2>, . . . <action m>."

A rule that controls traffic might have the following format:

```
(Rule traffic-signal
   (change-light red green)
   (light red)
   (car stop)
   (signal walk)
=>
   (retract (light red))
   (assert (light green))
   (retract (car stop))
   (assert (car go))
   (retract (signal walk))
   (assert (signal don't-walk))
); end traffic-signal.
```

Before the rule `traffic-signal` can fire, the following facts must exist in the knowledge base:

```
(change-light red green)
```

```
(light red)
(car stop)
(signal walk).
```

After the `traffic-signal` rule fires, the following facts will be in the knowledge base:

```
(change-light red green)
(light green)
(car go)
(signal don't-walk).
```

6.4.2 Strategies for Matching

6.4.2.1 Matching rules with facts

It is the purpose of an inference engine to manage the process of matching rules with facts. There are several approaches for pattern matching facts with rules. A simplistic approach is to have rules search for facts[1]. Figure 6-6 illustrates the rules-search-for-facts matching process graphically. The advantage of this method is that the computer programming required is simple to understand and the execution process is a straightforward, sequential activity. The disadvantage is that this method suffers from combinatorial explosion when the number of rules, rule patterns and facts increase. Under this matching scheme, about 90% of the total system execution time is spent in searching for facts that satisfy the rules. The balance of the processing time is spent determining which match should be acted on first (a process called conflict resolution) and completing the actions specified by the matched rules. It is desirable to significantly decrease the amount of time spent in the match phase and thereby increase the efficiency of the knowledge base system.

---

1. The discussion in the next two sections is based on lecture notes from the University of Florida course CAP 6627, Expert Systems, as taught by Dr. Douglas Dankel, II, Spring term, 1993.

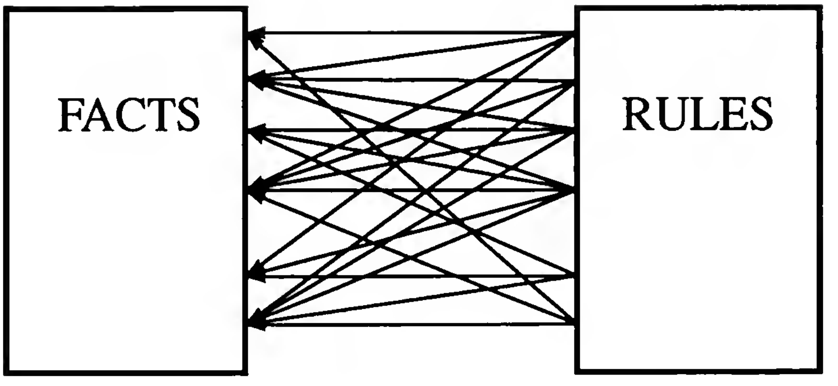


Figure 6-6. Matching rules with facts using a rules-search-for-facts approach.

The following scenarios show how quickly this method runs into the combinatorial explosion. The worst case number of searches required to find a single rule/fact pattern match can be expressed as

$$N_{search} = N_{rules} \times M_{cond} \times P_{facts} \tag{6.1}$$

where

N_{search} = number of pattern match comparisons
N_{rules} = number of rules
M_{cond} = average number of patterns per rule
P_{facts} = number of facts in the knowledge base.

Several numerical examples of the combinatorial explosion are summarized in Table 6-3. Notice that for the first case, which is unrealistic since it only has one pattern per rule, it may require as many as 10,000 comparisons to find a single rule/fact match. The third case, representative of a large system, requires up to 3,000,000 comparisons to find a suit-

able rule/fact match. High-speed, expensive computing hardware cannot practically make up for the slow response time in such an inefficient system.

Table 6-3. Combinatorial explosion for the rules-search-for-facts method.

N_{rules}	M_{cond}	P_{facts}	N_{search}
100	1	100	10^4
100	3	100	$3*10^4$
1000	3	1000	$3*10^6$

6.4.2.2 Exploiting temporal redundancy

Fortunately, only a small number of facts change with each cycle. In a system with hundreds of facts, less than one percent of the facts are likely to change with each cycle. This means that there is a slow rate of change in the conflict set. Since there is little change from one time period to the next, there is a large amount of redundant data. This condition is known as temporal redundancy and it can be exploited by having the facts search for rules rather than the other way around. Figure 6-7 illustrates how this approach can avoid unnecessary computation. This facts-match-rules strategy is clearly less cumbersome than the rules-match-facts strategy. The Rete algorithm was developed specifically to take advantage of this strategy.

6.4.2.3 Rete network

The Rete algorithm is an efficient many-many match procedure. The method involves the development of a pattern network and a join network. These networks are illustrated in Figure 6-8 for a sample rule. The Rete gains efficiency from reusing as many patterns from similar rules as possible. An example of two rules instantiated within a Rete network is shown in Figure 6-9. It can be seen from this figure how the pattern network reuses as much information from similar rules as possible. As facts enter the knowledge base, they are pushed as far as possible through the Rete networks. Facts that satisfy the patterns of

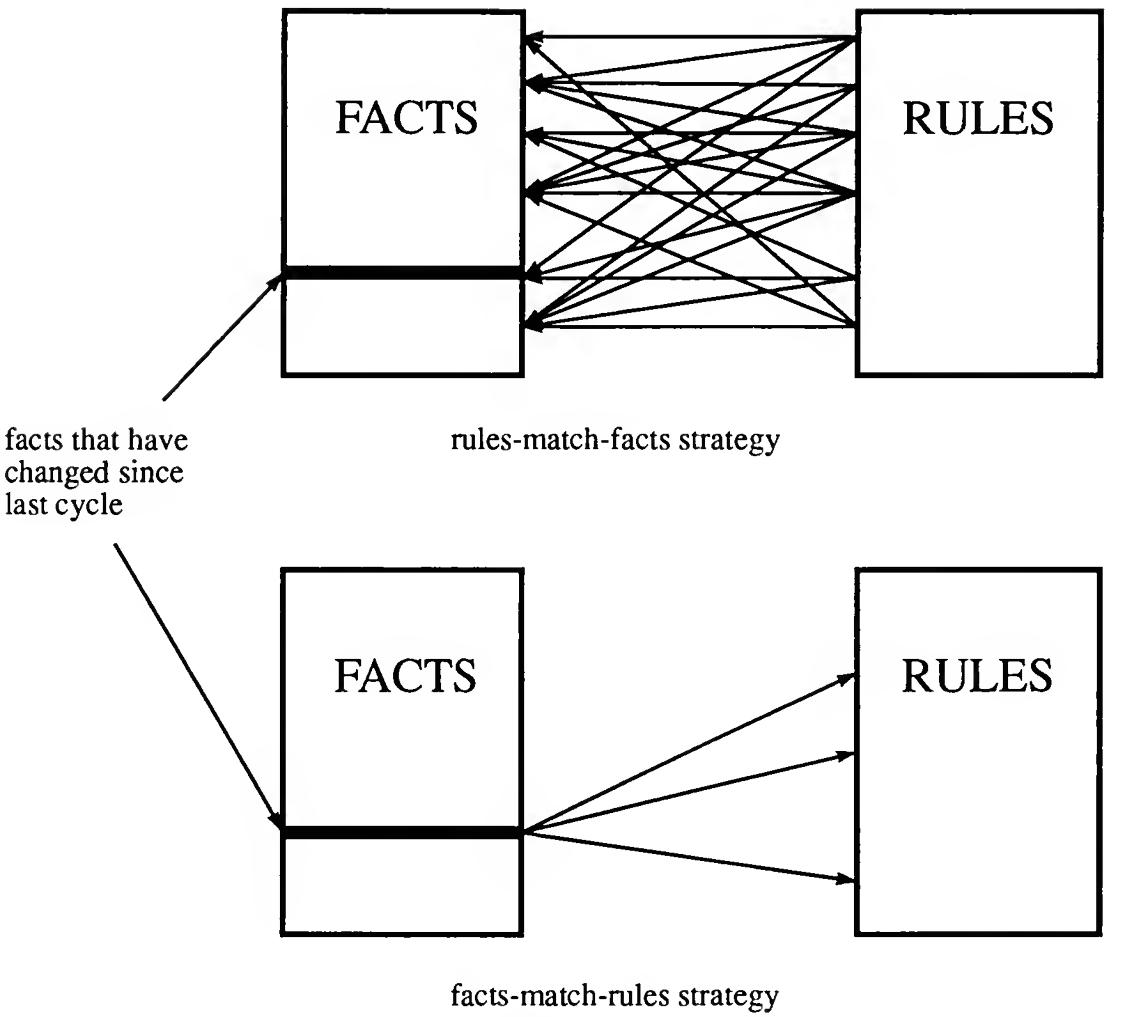


Figure 6-7. Comparison of two strategies for matching rules with facts.

the instantiated rules are placed on an agenda where the inference engine decides which matches to act on first.

6.5 CLIPS Expert System Shell

CLIPS[2] [Giar91, Giar93] is an expert system tool developed by the Software Technology Branch (STB), NASA/Lyndon B. Johnson Space Center. CLIPS is designed to facilitate the development of software to model human knowledge or expertise.

2. C Language Implementation Production System

<u>Sample rule</u>:

```
(rule R-1
   (field-1 field-2 field-3)
   (field-4 field-5)
      ...
=>
   ...); end rule R-1
```

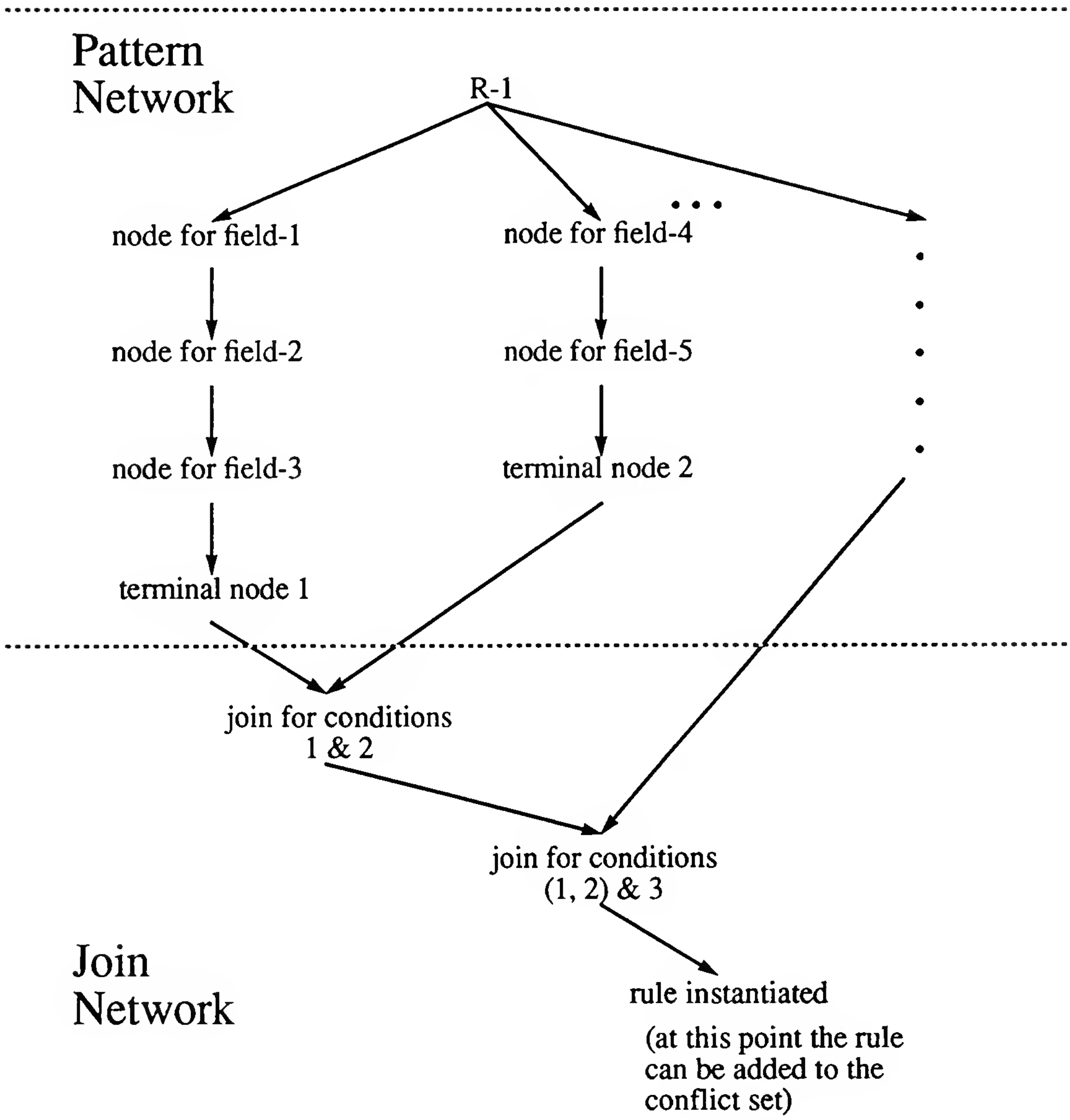


Figure 6-8. Instantiation of a typical rule within a Rete network.

Sample rules:

```
(rule R-2                       (rule R-3
   (T1 B ?D)                       (T1 B ?D)
   (T2 ?F)                         (T2 I)
   (T3 ?D ?F)                      (T3 ?D ?F)
=>                              =>
   ...); end rule R-2              ...); end rule R-3
```

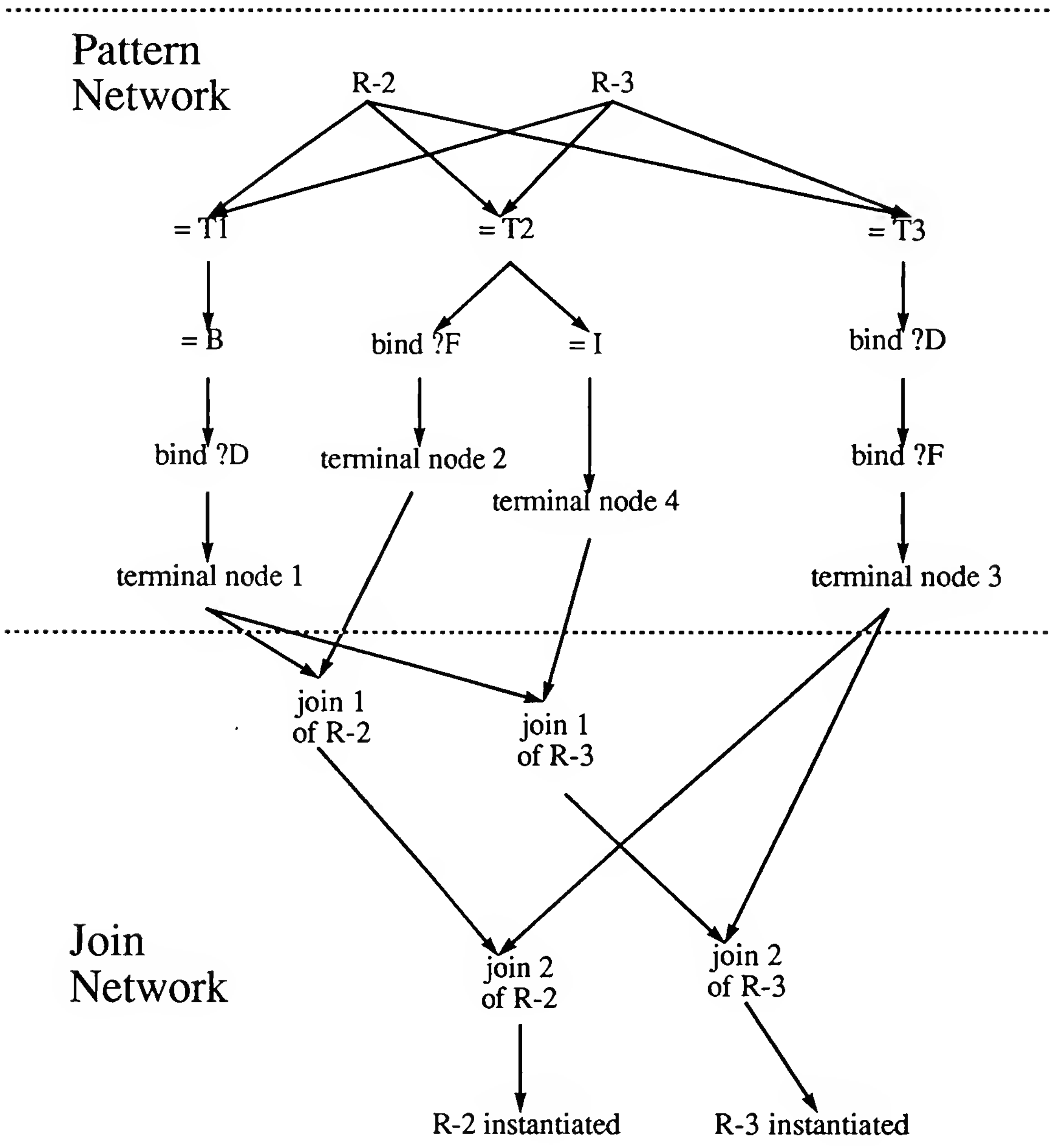


Figure 6-9. Instantiation of two rules within a Rete network.

There are three ways to represent knowledge in CLIPS:

- Rules, which are primarily intended for heuristic knowledge based on experience.
- Deffunctions and generic functions, which are primarily intended for procedural knowledge.
- Object-oriented programming[3], also primarily intended for procedural knowledge. The five generally accepted features of object-oriented programming are supported: classes, message-handlers, abstraction, encapsulation, inheritance, polymorphism. Rules may pattern match on objects and facts.

Software can be developed using only rules, only objects, or a mixture of objects and rules. Figure 6-10 shows the integrated CLIPS development environment as it is implemented on Apple Macintosh, and explains the CLIPS application windows. Figure 6-11 illustrates the CLIPS environment with a sample knowledge base source file being edited. The CLIPS application menus are briefly explained in Figure 6-11.

6.5.1 Defining Classes

Classes are defined within the CLIPS system by the `defclass` construct. A `defclass` is made up of the following five elements:

1. a name,
2. a list of parent classes, or superclasses, from which the new class inherits properties (slots) and message-handlers,
3. a specifier denoting whether or not direct instances of the new class may be created,
4. a specifier denoting whether or not instances of the class may be pattern matched with rules and
5. a list of slots specific to the new class.

6.5.1.1 The defclass construct

The syntax of the defclass construct is shown in Figure 6-12 and is explained in detail in the *CLIPS Reference Manual* [Giar93].

---

3. CLIPS Object-Oriented Language (COOL)

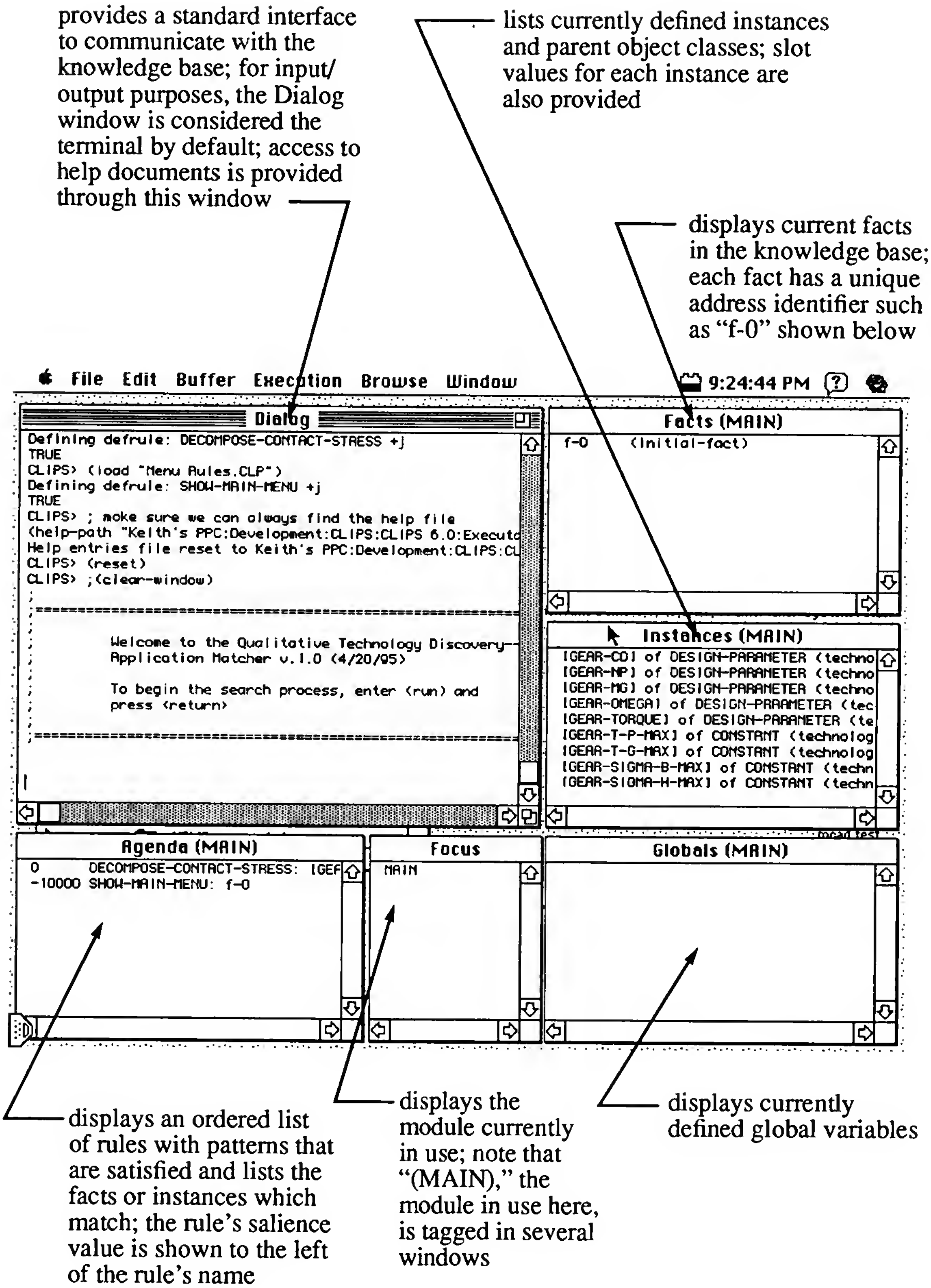


Figure 6-10. CLIPS knowledge base development system multiple document interface as implemented on the Macintosh computing platform.

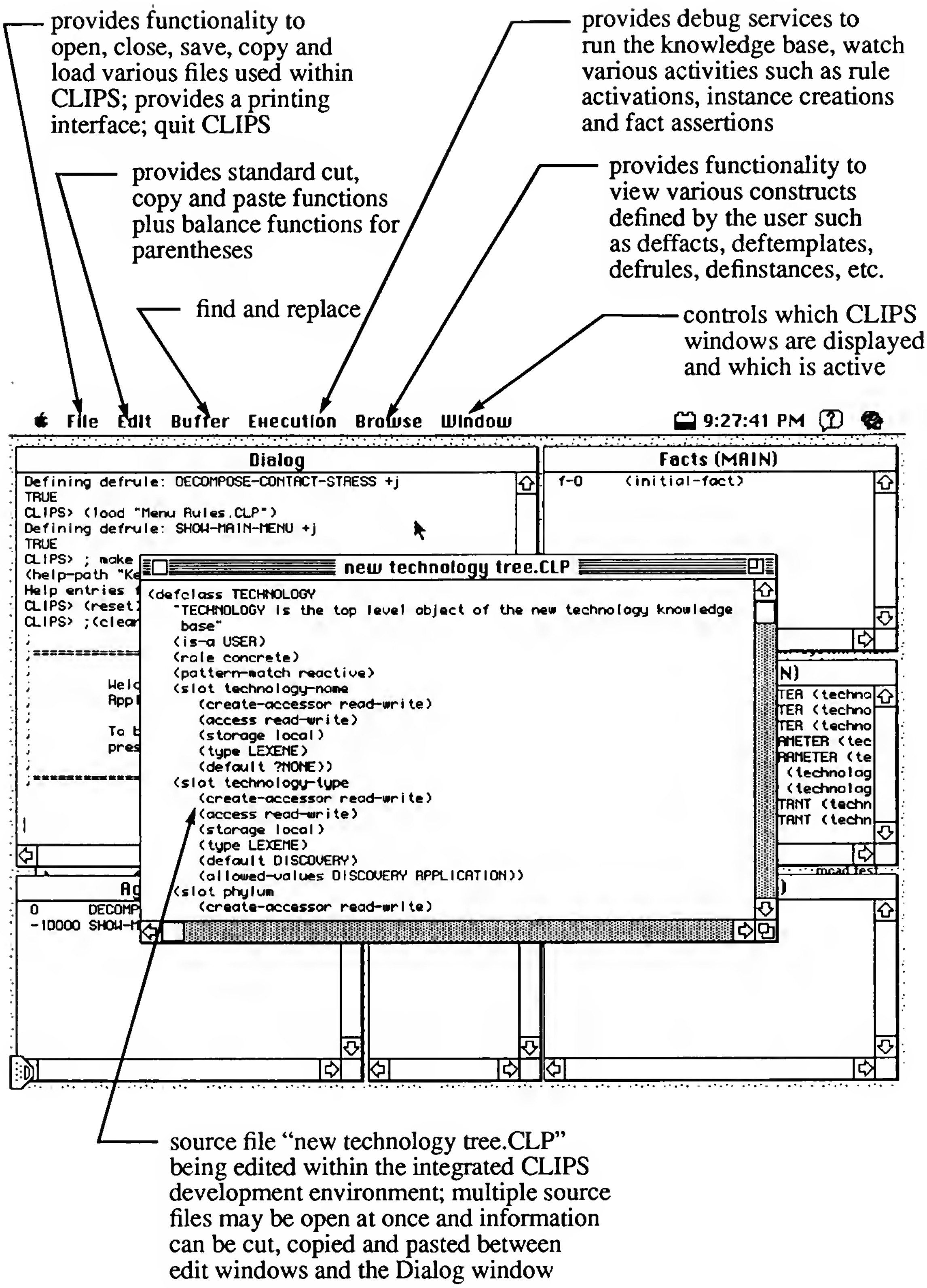


Figure 6-11. Edit of CLIPS source file and explanation of CLIPS application menus.

```
(defclass <name> [<comment>]
  (is-a <superclass-name>+)
  [<role>]
  [<pattern-match-role>]
  <slot>*
  <handler-documentation>*)

<role>  ::= (role concrete | abstract)

<pattern-match-role>
        ::= (pattern-match reactive | non-reactive)

<slot>  ::= (slot <name> <facet>*) |
            (single-slot <name> <facet>*) |
            (multislot <name> <facet>*)

<facet> ::=  <default-facet> | <storage-facet> |
             <access-facet> | <propagation-facet> |
             <source-facet> | <pattern-match-facet> |
             <visibility-facet> | <create-accessor-facet>
             <override-message-facet> | <constraint-attributes>

<default-facet> ::=
           (default ?DERIVE | ?NONE | <expression>*) |
           (default-dynamic <expression>*)

<storage-facet> ::= (storage local | shared)

<access-facet>
       ::= (access read-write | read-only | initialize-only)

<propagation-facet> ::= (propagation inherit | no-inherit)

<source-facet> ::= (source exclusive | composite)

<pattern-match-facet>
       ::= (pattern-match reactive | non-reactive)
```

Figure 6-12. CLIPS defclass construct.

6.5.1.2 Sample CLIPS class definitions

The source code shown in Figure 6-13 is an excerpt from the technology knowledge

```
<visibility-facet> ::= (visibility private | public)

<create-accessor-facet>
     ::= (create-accessor ?NONE | read | write | read-write)

<override-message-facet>
     ::= (override-message ?DEFAULT | <message-name>)

<handler-documentation>
       ::= (message-handler <name> [<handler-type>])

<handler-type> ::= primary | around | before | after
```

Figure 6-12—continued

base definition. The top-level class TECHNOLOGY, the abstract LEVEL-1-ATTRIBUTE class and the MATERIAL-ATTRIBUTE class are partially defined. The inheritance for each class is denoted by the is-a identifier. Since the TECHNOLOGY class is at the top of the user-defined hierarchy, its parent class is the USER class, which has predefined message-handlers that propagate through to its subclasses by default. These message-handlers include functions to get and put values into slots and to print the contents of an instance of the class. Note that the MATERIAL-ATTRIBUTE class inherits from two parent classes: (is-a TECHNOLOGY LEVEL-1-ATTRIBUTE).

A complete listing of all the classes defined for this study can be found in the Appendix.

6.5.2 Defining Instances

CLIPS has two approaches for creating instances of classes defined in the technology tree. The first method is interactive within the CLIPS expert system shell. The second

```
(defclass TECHNOLOGY
   "TECHNOLOGY is the top level object of the new
    technology knowledge base"
   (is-a USER)
   (role concrete)
   (pattern-match reactive)
   (slot technology-name
      (type LEXEME)
      (default ?NONE))
   (slot technology-type
      (type LEXEME)
      (default DISCOVERY)
      (allowed-values DISCOVERY APPLICATION))
   (slot phylum
      (type LEXEME)
      (default ?NONE))  ;technology code
)  ;end defclass TECHNOLOGY
(defclass MATERIAL-ATTRIBUTE
   "MATERIAL-ATTRIBUTE is a first-level attribute
    specialization of the TECHNOLOGY attribute hierarchy"
   (is-a TECHNOLOGY LEVEL-1-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot classification
      (default MATERIAL))
)  ;end defclass MATERIAL
```

Figure 6-13. Definition of the TECHNOLOGY and MATERIAL-ATTRIBUTE classes.

method, which is preferred, defines a group of instances in a source file outside the CLIPS environment that is then read by CLIPS to create the instances.

6.5.2.1 The make-instance construct

Instance definitions may be entered directly at the CLIPS command prompt. It is important to define all the underlying class structure prior to making instances. If this order is not followed, CLIPS reports an error. The syntax for creating an instance is shown in Figure 6-14. Note that information within the square brackets is optional. In this case, if

```
(make-instance <instance-definition>)

<instance-definition> ::= [<instance-name-expression>] of
                  <class-name-expression>
                  <slot-override>*
<slot-override>       ::= (<slot-name-expression>
<expression>*)
```

Figure 6-14. CLIPS make-instance construct.

the instance-name is left off, CLIPS assigns a unique value automatically. When an instance is entered properly, CLIPS returns the name of the instance as a confirmation. Figure 6-15 shows the results of entering three instances into the knowledge base. Note that the third `make-instance` was unsuccessful in this example.

The weakness of creating instances with the `make-instance` construct interactively is that once the CLIPS environment is reset, all facts and instances are cleared and must be reentered if needed. If the user wishes to have the instances created automatically after each reset, then the `definstances` construct, described below, must be used.

The best use of the `make-instance` construct is within the action portion of a rule. When used in this way, instances can be created without any intervention from the user. The `make-instance` construct is used in the action portion of the rule `MATCH-REQT-CAPABILITY`. The source code listing, found in the Appendix, contains the text for the rule `MATCH-REQT-CAPABILITY`.

6.5.2.2 The definstances construct

Each time a CLIPS knowledge base is reset, all facts and instances are deleted from the application's working memory. After a reset, it is desirable to have instances for specific classes be created. The `definstances` construct within CLIPS performs just that task. As with the `make-instance` construct, an error is reported if the user attempts to

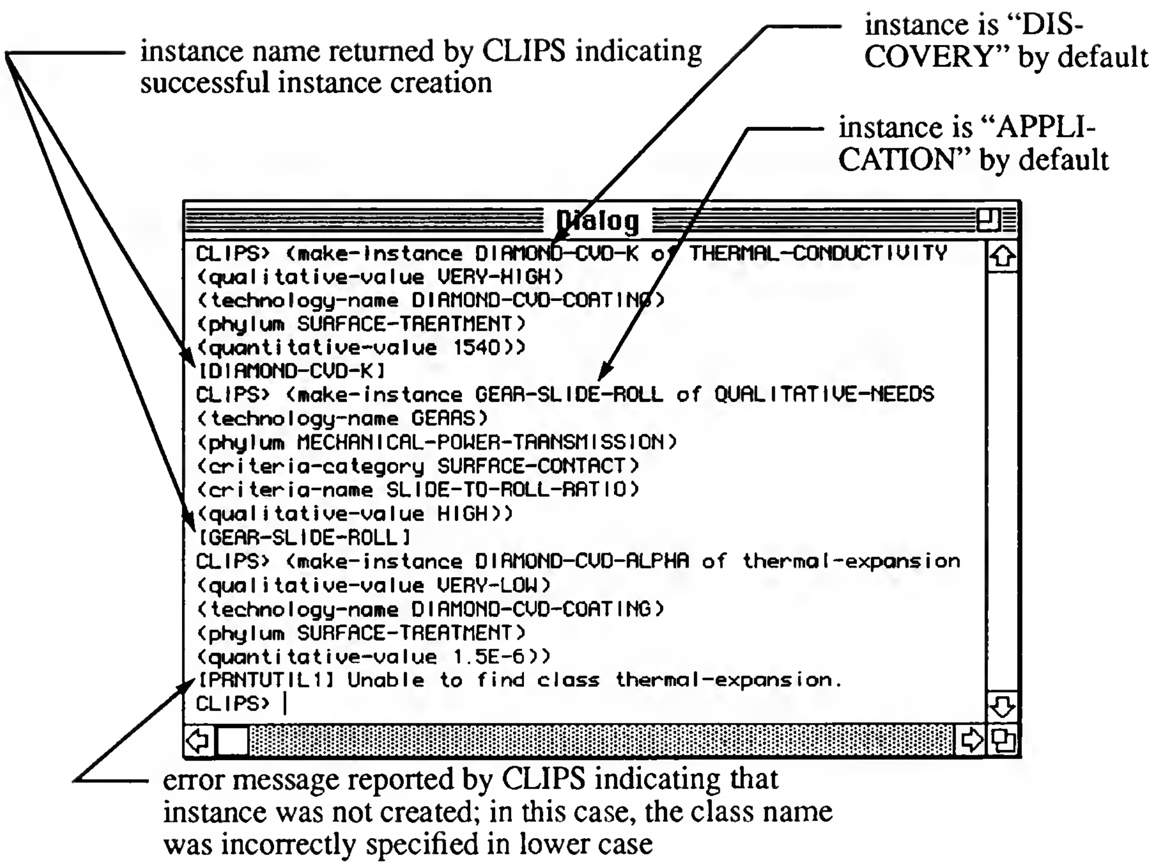


Figure 6-15. Interactive creation of knowledge base DISCOVERY instances in the CLIPS environment.

create an instance of a class before the class has been defined. Therefore, it is important to always create the structure (classes) first and the objects (instances) last.

The syntax for creating instances using the `definstances` construct is defined Figure 6-16.

CLIPS identifies a successfully defined definstance by returning "TRUE." An appropriate syntax error results if the definstance is improperly defined.

6.5.2.3 Sample CLIPS instance definition

Classes define underlying structure and inheritance links. Classes cannot contain data and therefore cannot be manipulated through pattern-matching or queries. Instances, however, are objects based on class definitions and can hold information that can be pattern-

```
(definstances <definstances-name> [active] [<comment>]
   <instance-template>*)
<instance-template> ::= (<instance-definition>)
<instance-definition> ::= <instance-name-expression> of
                              <class-name-expression>
                              <slot-override>*
<slot-override>        ::= (<slot-name-expression>
<expression>*)
```

Figure 6-16. CLIPS definstances construct.

matched or manipulated through queries. An instance inherits its structure from the class and may also derive default slot values. In the previous code fragment, the `classification` slot has a default value of `MATERIAL`. An instance is based on the `THERMAL-CONDUCTIVITY` class is shown in Figure 6-17. The `THERMAL-CONDUCTIVITY` class

```
(make-instance DIAMOND-CVD-K of THERMAL-CONDUCTIVITY
   (qualitative-value VERY-HIGH)
   (technology-name DIAMOND-CVD-COATING)
   (phylum SURFACE-TREATMENT)
   (quantitative-value 1540)
   )  ; end instance DIAMOND-CVD-K.
```

Figure 6-17. Definition of the DIAMOND-CVD-K instance.

definition is provided in Figure 6-18. Results of a print message sent to the `DIAMOND-CVD-K` instance are illustrated in Figure 6-19.

It is interesting to note that although the make-instance only provided values for four slots, the default values for the base class `THERMAL-CONDUCTIVITY` provided information for the remaining slots. This example illustrates the advantages of data abstraction and inheritance within the OOP paradigm. A complete set of technology class templates

```
(defclass THERMAL-CONDUCTIVITY
   "THERMAL-CONDUCTIVITY is a FOURTH-level attribute
    specialization of the TECHNOLOGY attribute hierarchy under
    the THERMAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY THERMAL-PROPERTY
          LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default HEAT-TRANSFER))
   (slot units
      (default Watt/meter-deg-K))
   (slot symbol
      (default K))
) ;end defclass THERMAL-CONDUCTIVITY
```

Figure 6-18. Definition of the THERMAL-CONDUCTIVITY class.

are provided in the Appendix. In the next chapter, a step-by-step procedure for using the templates to create APPLICATION and DISCOVERY instances are presented. Figure 6-20 shows a source file as displayed in the integrated CLIPS environment's editor.

6.5.3 Defining Rules

Facts and instances in the CLIPS system match on patterns defined in rules according to the results of the Rete network activity. Rules fire in an order determined by how successful matches are placed on the agenda. As rules fire, the state of the knowledge base changes—meaning that certain facts and instances are added, deleted or modified. The agenda is updated after each cycle to determine the next rule to fire.

6.5.3.1 The defrule construct

Rules are defined in CLIPS with the `defrule` construct. The syntax for the `defrule` construct is provided in Figure 6-21. Note that the `<conditional-element>` is a set of fact and/or instance patterns for matching and the `<action>` is a group of consequent actions performed on firing.

```
CLIPS> (send [DIAMOND-CVD-K] print)
[DIAMOND-CVD-K] of THERMAL-CONDUCTIVITY
(assessment-criteria CAPABILITY)
(technology-name DIAMOND-CVD-COATING)
(technology-type DISCOVERY)
(phylum SURFACE-TREATMENT)
(classification MATERIAL)
(order PHYSICAL)
(family THERMAL)
(qualitative-value VERY-HIGH)
(qualitative-value-range nil)
(quantitative-value 1540)
(quantitative-value-range 0)
(species HEAT-TRANSFER)
(units Watt/meter-deg-K)
(symbol K)
CLIPS>
```

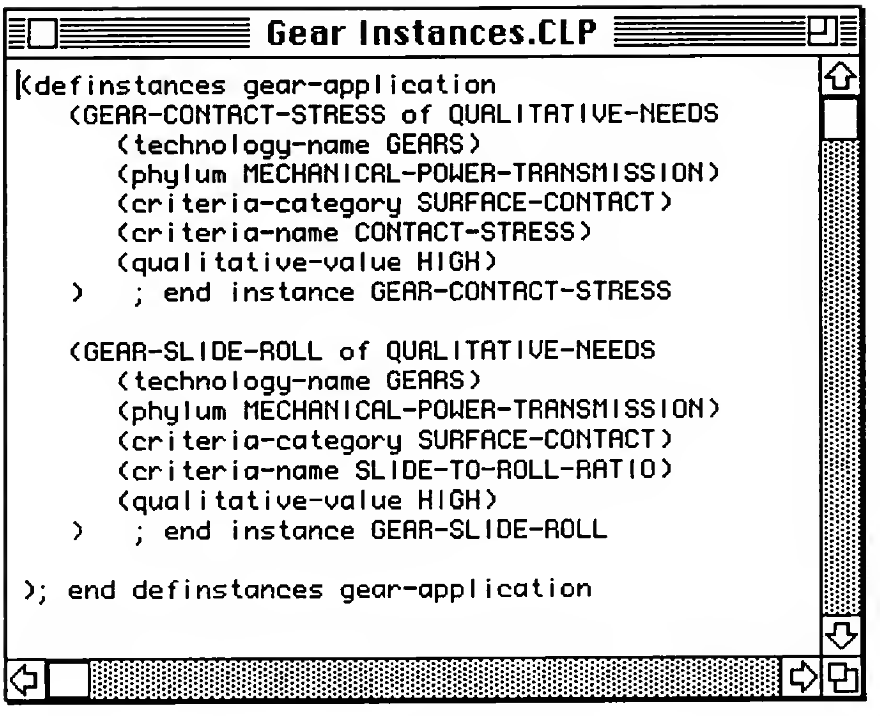
Figure 6-19. Results of a print message sent to the DIAMOND-CVD-K instance.

```
(defrule <rule-name> [<comment>]
    [<declaration>]
    <conditional-element>*
    =>
    <action>*)
```

Figure 6-21. CLIPS defrule construct.

6.5.3.2 Sample CLIPS rule definition

The rule shown in Figure 6-22 searches for a DISCOVERY instance and an APPLICATION instance that have the same `species`. If a match is found, an instance of the `MATCH-TABLE` class is created. The "goodness" of fit for the match is determined by the results of the `match-rating` function.




```
(definstances gear-application
   (GEAR-CONTACT-STRESS of QUALITATIVE-NEEDS
      (technology-name GEARS)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (criteria-category SURFACE-CONTACT)
      (criteria-name CONTACT-STRESS)
      (qualitative-value HIGH)
   )   ; end instance GEAR-CONTACT-STRESS

   (GEAR-SLIDE-ROLL of QUALITATIVE-NEEDS
      (technology-name GEARS)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (criteria-category SURFACE-CONTACT)
      (criteria-name SLIDE-TO-ROLL-RATIO)
      (qualitative-value HIGH)
   )   ; end instance GEAR-SLIDE-ROLL

); end definstances gear-application
```

Figure 6-20. CLIPS integrated editor window showing example of an external source file with a `definstances` construct.

6.5.4 Defining Knowledge Base Queries

CLIPS provides a mechanism for querying the knowledge base much like a database user would query a relational database. Queries reduce some of the pattern-matching overhead associated with the Rete network. The CLIPS query system is implemented in the COOL language. The query system is useful for determining and performing actions on sets of instances of user-defined classes that satisfy user-defined queries. COOL provides six functions which each operate on sets of instances as determined by user-defined criteria. A description of the available query functions is given in Table 6-4.

6.5.4.1 The query structure

The instance-set query function used most often in the reported study's knowledge base system is `do-for-all-instances`. The syntax of this function is provided in

```
(defrule MATCH-REQT-CAPABILITY
   "This rule builds a table of instances for DISCOVERY
    capabilities that match APPLICATION requirements. The
    matching table has the following attributes for the
    preliminary ranking process:
      discovery-name
      application-name
      criterion-matched
      discovery-qualitative-value
      application-qualitative-value
      match-goodness
      match-id
      discovery-attribute-address
      application-attribute-address
   "
   ?appl <- (object (is-a ?x)
                    (species ?spec)
                    (technology-type APPLICATION))
   ?disc <- (object (is-a ?x)
                    (species ?spec)
                    (technology-type DISCOVERY))
=>
   (make-instance of MATCH-TABLE
      (discovery-attribute-address ?disc)
      (application-attribute-address ?appl)
      (discovery-name (send ?disc get-technology-name))
      (application-name (send ?appl get-technology-name))
      (criterion-matched (class ?appl))
      (discovery-qualitative-value (send ?disc
         get-qualitative-value))
      (application-qualitative-value (send ?appl
         get-qualitative-value))
      (match-goodness (match-rating (send ?appl
         get-qualitative-value)
                 (send ?disc get-qualitative-value)
                 (send ?appl get-assessment-criteria)))
   ) ;end make-instance MATCH-TABLE
) ;end defrule MATCH-REQT-CAPABILITY
```

Figure 6-22. Definition of the MATCH-REQT-CAPABILITY rule.

Table 6-4. COOL query system functions.

Function	Purpose
any-instancep	Determines if one or more instance-sets satisfy a query
find-instance	Returns the first instance-set that satisfies a query
find-all-instances	Groups and returns all instance-sets which satisfy a query
do-for-instance	Performs an action for the first instance-set which satisfies a query
do-for-all-instances	Performs an action for every instance-set which satisfies a query as they are found
delayed-do-for-all-instances	Groups all instance-sets which satisfy a query and then iterates an action over this group

Figure 6-23. A detailed discussion of instance-sets is provided in the *CLIPS Reference Manual* [Giar93].

```
(do-for-all-instances <instance-set-template> <query> <action>)
<instance-set-template>
      ::= (<instance-set-member-template>+)
<instance-set-member-template>
      ::= (<instance-set-member-variable> <class-restrictions>)
<instance-set-member-variable> ::= <single-field-variable>
<class-restrictions>           ::= <class-name-expression>
<query> ::= <boolean-expression>
```

Figure 6-23. CLIPS do-for-all-instances query construct.

6.5.4.2 Sample CLIPS knowledge base query

Queries may be executed directly from the CLIPS command prompt or built into functions or rules. Two sample queries that export all of the equations for a technology match are shown in Figure 6-24. Notice how the queries are built into the action side of the rule.

6.5.5 Loading CLIPS Constructs

CLIPS provides a convenient mechanism to import constructs from an external source file. The `load` command is used at the CLIPS prompt to import externally defined con-

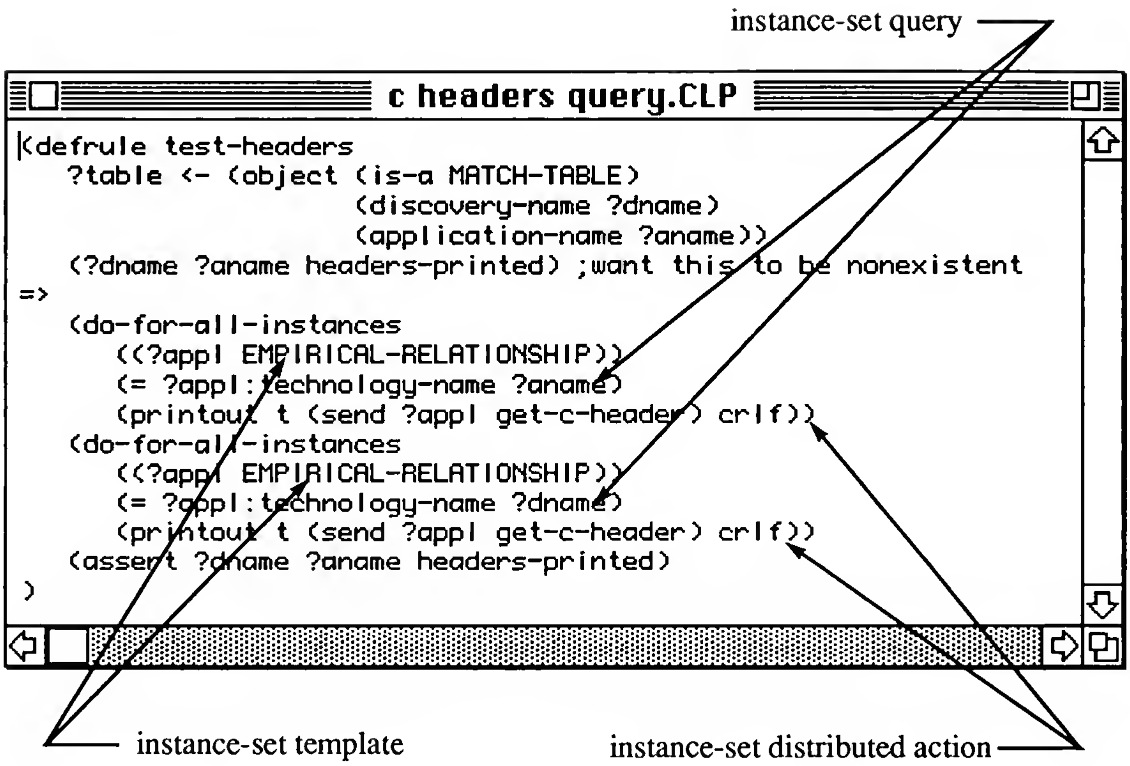


Figure 6-24. Instance-set queries built into the action of a rule.

structs such as `definstances` and `deffacts` (a way to create facts that are recreated after each reset of the environment). The syntax for the `load` command is as follows: `(load <file-name>)`. Figure 6-25 shows the result of a typical `load` command.

It is also useful to perform a series of external source file loads in a batch mode. The CLIPS `batch` command provides this capability. The syntax for the `batch` command is as follows: `(batch <file-name>)`. The `batch` function executes commands one at a time from a source file. If not ordered properly the batch file can result in many errors. The best way to avoid problems is to check each action in a batch file using the `load` command first.

The following list is a suggested order for constructs arranged within a batch file:

1. defglobals,
2. deffacts/deftemplates,

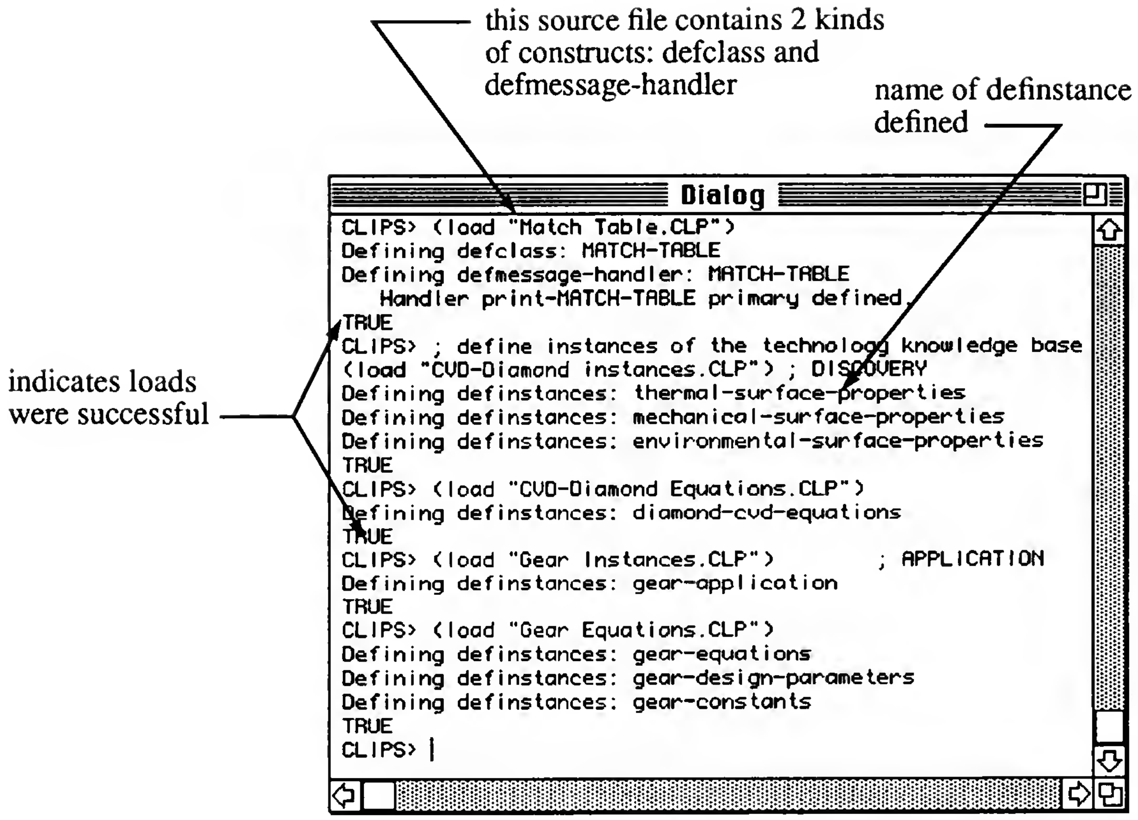


Figure 6-25. Sample external source files successfully loaded into the knowledge base.

3. defgenerics,
4. deffunctions,
5. defclasses,
6. defmessage-handlers,
7. definstances and
8. defrules

Figure 6-26 illustrates a typical batch file in action.

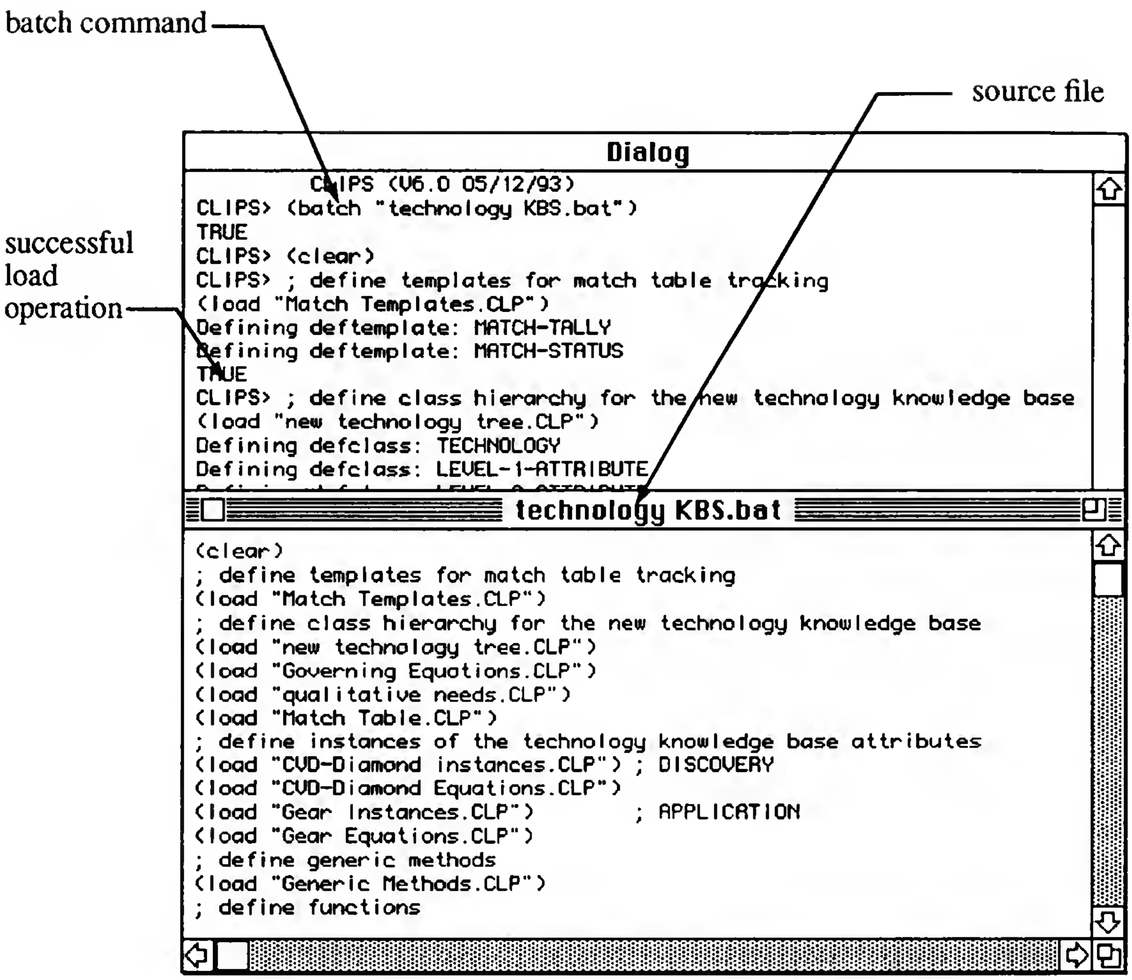


Figure 6-26. Batch file and sample execution of the batch command within the CLIPS environment.

CHAPTER 7
APPLICATION OF THE QUALITATIVE MATCHING STRATEGY

Chapter 6 examined the underlying structure of the qualitative matching process. A technology class hierarchy and the rules used in performing the matching were defined. This chapter presents a step-by-step procedure for creating the DISCOVERY and APPLICATION knowledge bases. Specific instances of the technologies developed in the previous chapters, namely the diamond CVD coating process (the DISCOVERY knowledge base) and gear systems (the APPLICATION knowledge base), illustrate the process. A sample execution of the qualitative matching procedure completes this chapter. The results of this process are used in Chapter 8 to quantitatively analyze the qualitative matches for cost/benefit and ranking.

Much of the information presented in this chapter is formatted in a user's manual style to promote the customizing the process for other technology domains.

7.1 The Implementation Steps of the Developed Process

The developed qualitative matching process is implemented in a prototype application system. The application framework consists of technology classes, matching classes, matching rules and procedural functions. Instances of the technology classes and qualitative rules are defined to populate the knowledge base management application system. Execution results are post-processed by the quantitative procedures presented in the next chapter.

7.1.1 Construction of the Prototype Application

The prototype application was constructed in three phases. First, the class structure of the technology knowledge bases was developed and defined. Next, qualitative matching

rules were developed. Last, supporting procedural functions were implemented. These functions provide services such as the menu system and output capabilities.

7.1.1.1 Definition of Classes

Several types of classes were designed to structure information in the knowledge base application system. Technology classes provide an underlying structure for discovery and application technology instances. The matching classes provide a mechanism for holding statistics on discovery-application matched pairs.

Technology classes. The attribute hierarchy developed in the previous chapter was encoded in a series of source files for incorporation within the application system. The complete listing of all the source files and technology class structures is provided in the Appendix.

Matching classes. Two class structures were defined to gather information regarding discovery-application attribute matches. A match table structure (the defclass MATCH-TABLE) was defined to hold specific information on each attribute match. The table contains the names of the discovery and application technologies, the attribute name and the match rating. The match rating for a given attribute match is either "strong" or "weak" depending on how closely the qualitative values compare. The MATCH-TABLE structure allows the system user to review match instances individually and assess the match validity.

The other match class structure (the defclass MATCH-TALLY) maintains a running total of strong and weak matches for each discovery-application pair. MATCH-TALLY is used for presenting final matching statistics.

7.1.1.2 Definition of matching rules

The qualitative matching process works by comparing discovery technology attributes with potential application technology attributes on the atomic or individual attribute level. Attributes can be defined directly at the atomic level or can be indirectly defined using qualitative requirements. The advantage to indirect attribute definition is that the potential

application qualitative needs can be defined at a higher level of abstraction. The disadvantage is that intermediate decomposition rules must be written to transform qualitative requirements into atomic attributes.

Atomic attribute matching. The rule MATCH-REQT-CAPABILITY was developed to handle atomic attribute matching of discovery-application pairs. For each application attribute instance species that matches a given discovery attribute instance species, an instance in the match table is created. Each match is rated for goodness of fit (either "strong" or "weak") with rating stored in the match table "match-goodness" slot.

Qualitative requirements decomposition. DECOMPOSE-CONTACT-STRESS is a sample rule that breaks high-level qualitative requirements down into individual atomic attributes for use in the qualitative matching process.

7.1.1.3 Definition of rules and functions to support the application framework

The final structural element in the application system framework is the rule, Main-Menu, that fires when all the matching completes. This rule uses a series of procedural functions that display menus allowing the system user to post-process matching results. A complete set of the framework support functions, encoded in the source file "Menu System.CLP," is listed in the Appendix.

7.1.2 Populating the Knowledge Bases

Once all the structural elements of the qualitative matching system are encoded, the technology knowledge bases can be populated. For a given technology, instances of appropriate classes are defined and loaded into the application system. A road map of the procedure is shown in Figure 7-1. The process is defined in detail later in this chapter for one sample discovery technology and two sample potential application technologies.

7.1.3 Executing the System

Details on executing the system to determine potential application matches for discovery technologies are detailed later in this chapter under the section "Sample Execution of the Qualitative Matching Engine."

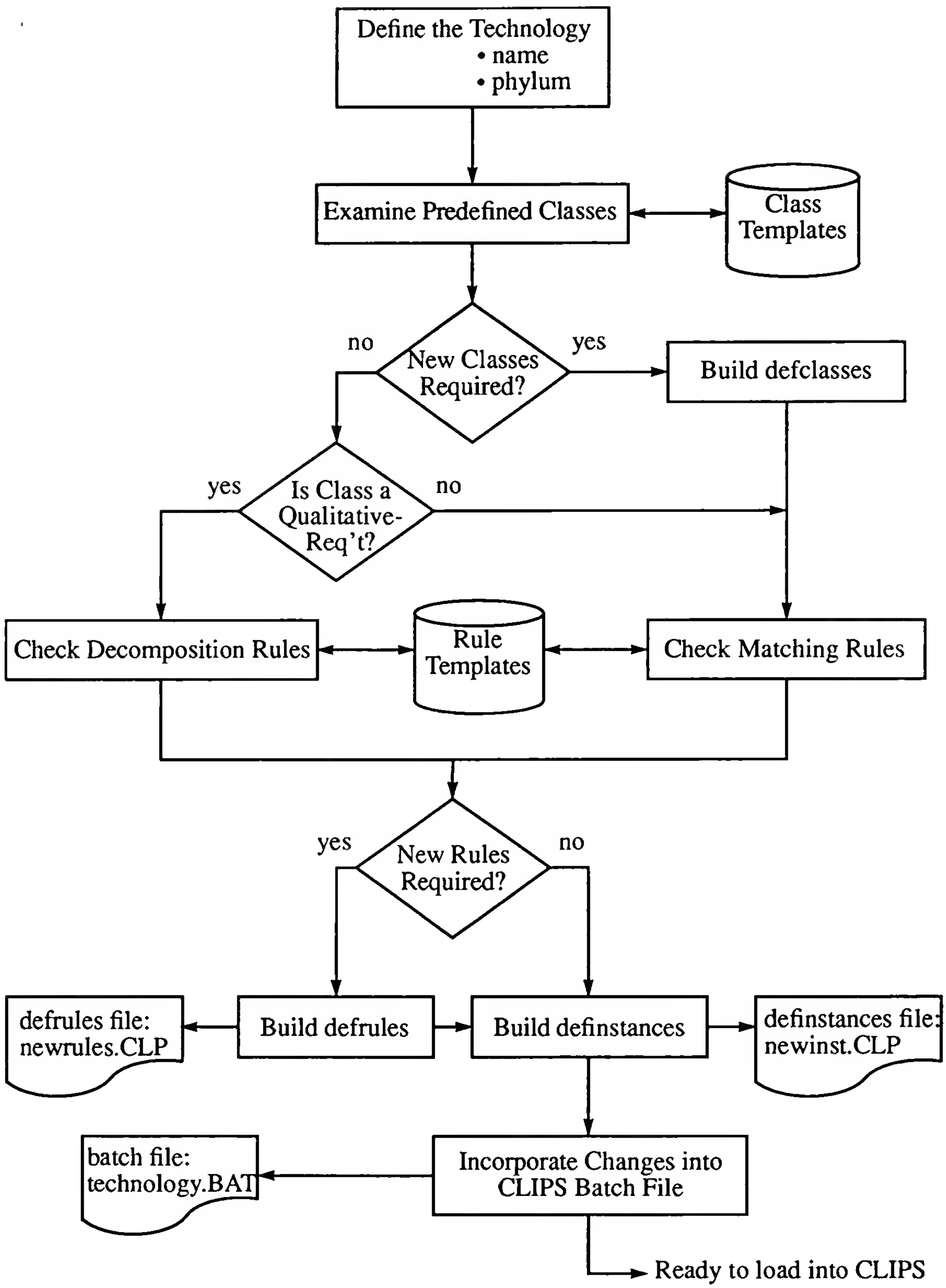


Figure 7-1. Process for building knowledge base entities for a technology.

7.2 Building Instances of the Technology Classes

The DISCOVERY and APPLICATION knowledge bases are built by defining specific instances of the technology hierarchy attribute tree and incorporating rules specific to the their respective domains. Before building instances of the technology classes, it is important to determine a name for the technology, a phylum to classify the technology and naming conventions for instances of the technology's parent classes. After these factors are considered, the process of building specific DISCOVERY and APPLICATION instances becomes more apparent.

7.2.1 Decisions Common to DISCOVERY and APPLICATION Instances

Before a technology instance can be built, there are three preliminary questions to answer. First, what is the technology-name? Second, what phylum does the technology belong to? Third, how should instances of the technology classes be named?

7.2.1.1 Technology-name selection

The technology-name slot provides a unique identifier for the instances of a particular technology. The CLIPS system is case sensitive. Therefore, capitalization must be consistent throughout the instance-building process. It is suggested that the technology-name be entered in all uppercase. Multiple words in the name must be separated by hyphens. For example, the DISCOVERY under consideration in this study, the CVD diamond coating process, is named "DIAMOND-CVD-COATING." The APPLICATION under consideration, standard dedendum gear pairs, is denoted as "GEARS." Shorter names save typing, but it is best to be more descriptive if more than one type of application within a technology is under consideration.

7.2.1.2 Phylum determination

The phylum slot identifies a broad category for classifying a technology. Table 7-1 provides examples of phylums along with applicable technologies. As with the technol-

Table 7-1. Sample phylums and associated technologies.

Phylum Name	Applicable Technologies
SURFACE-TREATMENT	thermal barrier coatings, plating, hardfacing, shot peening
MECHANICAL-POWER-TRANSMISSION	gear pairs, hydrostatic drives, traction drives
HEAT-DISSIPATION	heat sinks, brakes
ENERGY-STORAGE	flywheels, capacitors
BEARINGS	fluid-film bearings, dry-film bearings, rolling-element bearings

ogy-name slot, it is best to type phylum names in all uppercase. The phylum in use with the CVD-DIAMOND-COATING technology-name is "SURFACE-TREATMENT."

7.2.1.3 Instance naming guidelines

Each instance in a CLIPS knowledge base must be uniquely named. If a name is reused, then the previous instance is deleted. A useful guideline is to incorporate part of the technology-name plus a symbol name from the class into the instance names. For example, the instance, DIAMOND-CVD-K, which describes the thermal conductivity property for CVD diamond films, derives "DIAMOND-CVD" from the technology-name slot and "K" from the symbol slot.

Instances of classes where the symbol is uncommon, is nonexistent or includes subscripted characters, may be named using the class name or an abbreviation in place of the symbol value. For example, there is no symbol associated with the OXIDATION-RESISTANCE class. Therefore, a reasonable instance name is "DIAMOND-CVD-OX-RESIST."

7.2.2 Using the Technology Attribute Class Templates

The Appendix includes a complete set of technology attribute class templates. The templates list each slot defined for each class and the default values and allowed values for each slot. Table 7-2 is a sample template for the MELT-POINT class. These templates are

Table 7-2. MELT-POINT class template.

<table>
<tr><td colspan="4">Class Name: MELT-POINT
Parent Classes: TECHNOLOGY-PROPERTY, THERMAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description: first-level specialization of the THERMAL-PROPERTY class defining the melting point property; direct instances of this class can be created</td></tr>
<tr><th>Slot Name</th><th>Req'd?</th><th>Default Value</th><th>Allowed Values</th></tr>
<tr><td>technology-name</td><td>yes</td><td></td><td></td></tr>
<tr><td>technology-type</td><td></td><td>DISCOVERY</td><td>DISCOVERY|APPLICATION</td></tr>
<tr><td>phylum</td><td>yes</td><td></td><td></td></tr>
<tr><td>classification</td><td></td><td>MATERIAL</td><td></td></tr>
<tr><td>order</td><td></td><td>PHYSICAL</td><td></td></tr>
<tr><td>family</td><td></td><td>THERMAL</td><td></td></tr>
<tr><td>species</td><td></td><td>HEAT-TRANSFER</td><td></td></tr>
<tr><td>assessment-criteria</td><td>yes</td><td>CAPABILITY</td><td>NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY</td></tr>
<tr><td>qualitative-value</td><td></td><td></td><td>VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH</td></tr>
<tr><td>qualitative-value-range</td><td></td><td></td><td></td></tr>
<tr><td>quantitative-value</td><td></td><td></td><td></td></tr>
<tr><td>quantitative-value-range</td><td></td><td></td><td></td></tr>
<tr><td>units</td><td></td><td>deg-K</td><td></td></tr>
<tr><td>symbol</td><td></td><td>T-sub-melt</td><td></td></tr>
</table>

intended to be used as a guide for building technology class instances. A complete listing of class templates is available in the Appendix.

Figure 7-2 shows an instance of the MELT-POINT class as defined for the DIAMOND-CVD-COATING technology. The results of a print message sent to the instance are given in Figure 7-3. Table 7-3 lists all the slot values for the DIAMOND-CVD-MELT-POINT instance.

The suggested method of building technology class instances is as follows:

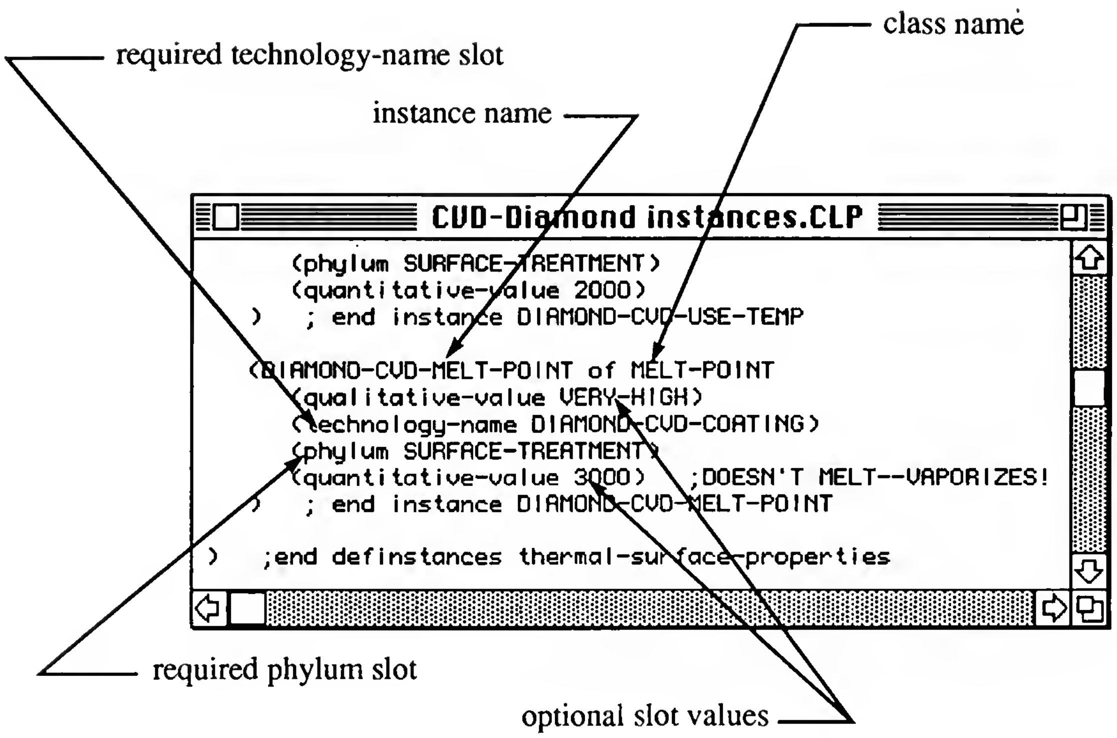


Figure 7-2. Definition of a MELT-POINT instance within a definstances construct.

1. gather properties, governing equations and other attribute information related to the technology under investigation,

2. consult the class templates in the Appendix to select the proper technology attribute classes,

3. assign a technology-name and phylum,

4. using the information gathered in step 1, assign values to as many slots as possible,

5. determine instance names for each attribute,

6. create a text source file and use the `definstances` construct to build the instances and

7. `load` the instances into the knowledge base (a batch file is the most convenient method to accomplish the loading procedure).

Table 7-4 is a template for gathering and organizing technology physical properties. This template can be used as a work sheet to build instances prior to editing a source file.

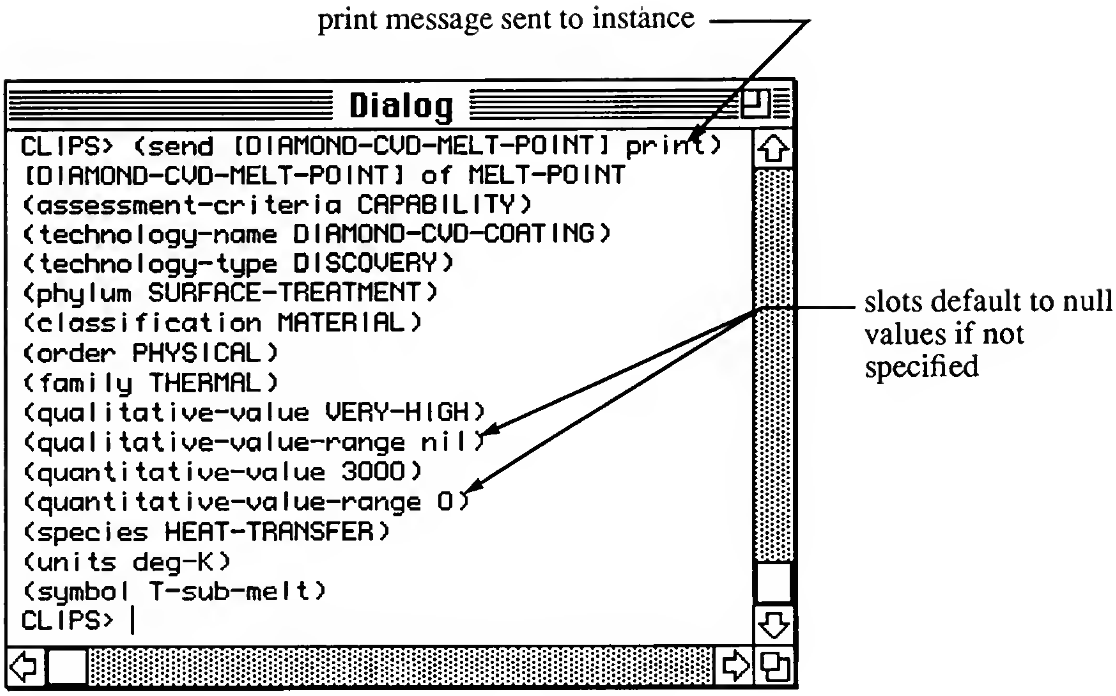


Figure 7-3. Resulting instance DIAMOND-CVD-MELT-POINT slot values following definstances invocation.

Table 7-3. DIAMOND-CVD-MELT-POINT instance slot values.

Instance Name: DIAMOND-CVD-MELT-POINT Class Name: MELT-POINT	
Slot Name	Value
technology-name	DIAMOND-CVD-COATING
technology-type	DISCOVERY
phylum	SURFACE-TREATMENT
classification	MATERIAL
order	PHYSICAL
family	THERMAL
species	HEAT-TRANSFER
assessment-criteria	CAPABILITY
qualitative-value	VERY-HIGH
qualitative-value-range	
quantitative-value	3000
quantitative-value-range	
units	deg-K
symbol	T-sub-melt

Table 7-4. Work sheet for gathering technology physical properties.

Technology Name:			Phylum:		
Property Name	Quantitative		Qualitative		Instance Name
	Value	Range	Value	Range	

Similar templates can be used for other types of technology attributes, such as governing equations and manufacturing properties. Figure 7-4 shows a `definstances` construct being built in a source file.

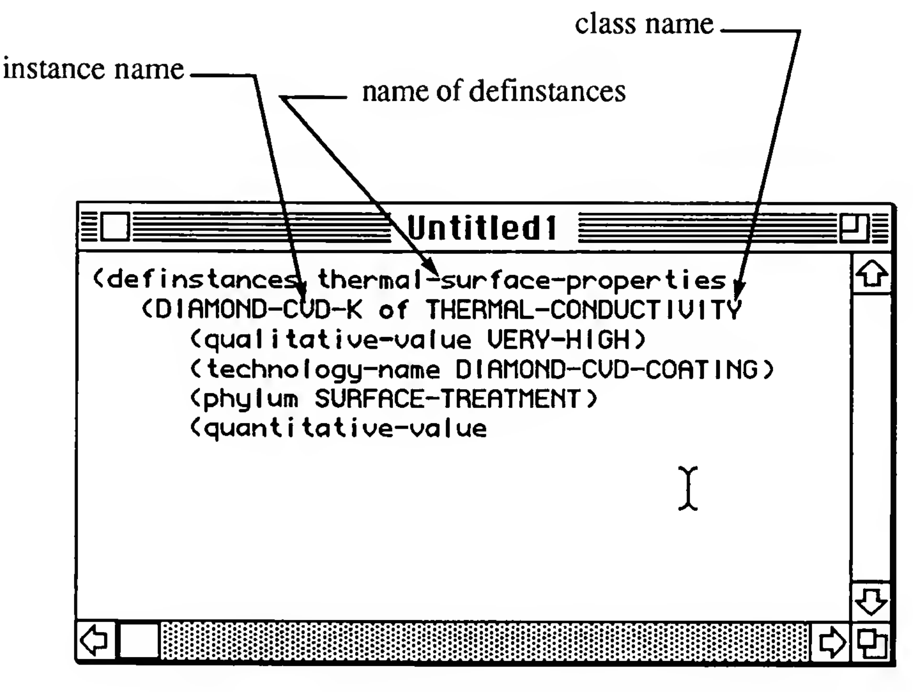


Figure 7-4. Source file with a definstances construct being built.

The next two sections illustrate specifically how to build the instances needed for the DIAMOND-CVD-COATING and GEARS knowledge base. The seven steps above guide the process.

7.2.3 DISCOVERY Instances

The step-by-step procedure for building instances of the technology classes for the discovered CVD diamond process is detailed in this section. Technology attributes in the form of physical properties and governing equations are gathered, organized and input into the knowledge base for use in the matching process.

7.2.3.1 Physical properties

Gather attribute information.Thermal properties for diamond were collected and entered into the work sheet shown in Table 7-5. Similarly, mechanical properties are given

Table 7-5. Work sheet for DIAMOND-CVD thermal-physical properties.

Technology Name: DIAMOND-CVD			Phylum: SURFACE-TREATMENT		
Property Name	Quantitative		Qualitative		Instance Name
	Value	Range	Value	Range	
thermal conductivity	1540		very high		DIAMOND-CVD-K
specific heat	853		medium		DIAMOND-CVD-C
thermal expansion coefficient	$1.5*10^{-6}$		very low		DIAMOND-CVD-ALPHA
thermal diffusivity	$1291*10^{-6}$		very high		DIAMOND-CVD-ALPHA-DIFF
maximum usage temperature	2000		very high		DIAMOND-CVD-USE-TEMP
melting point	3000		very high		DIAMOND-CVD-MELT-POINT

in Table 7-6 and environmental properties in Table 7-7.

Select appropriate class templates. The class templates, located in the Appendix, were reviewed and templates for the following classes were selected: THERMAL-CONDUCTIVITY, SPECIFIC-HEAT, THERMAL-EXPANSION, THERMAL-DIFFUSIVITY,

Table 7-6. Work sheet for DIAMOND-CVD mechanical-physical properties.

Technology Name: DIAMOND-CVD			Phylum: SURFACE-TREATMENT		
Property Name	Quantitative		Qualitative		Instance Name
	Value	Range	Value	Range	
Young's modulus	$1.05*10^{12}$		very high		DIAMOND-CVD-E
shear modulus	$5.5*10^{11}$		very high		DIAMOND-CVD-G
Poisson's ratio	0.07		very low		DIAMOND-CVD-NU
density	3500		medium low		DIAMOND-CVD-RHO
surface roughness	$0.1*10^{-6}$		medium low		DIAMOND-CVD-R
hardness (Knoop)	5700		very high		DIAMOND-CVD-HK

Table 7-7. Work sheet for DIAMOND-CVD environmental-physical properties.

Technology Name: DIAMOND-CVD			Phylum: SURFACE-TREATMENT		
Property Name	Quantitative		Qualitative		Instance Name
	Value	Range	Value	Range	
oxidation resistance			very high		DIAMOND-CVD-OX-RESIST
Property Name	List of Items				Instance Name
compatible lubricants	water, oil				DIAMOND-CVD-LUB-COMPAT

USAGE-TEMPERATURE, MELT-POINT, YOUNGS-MODULUS, SHEAR-MODULUS, POISSON-RATIO, DENSITY, ROUGHNESS, HARDNESS, OXIDATION-RESISTANCE and LUB-COMPATIBILITY.

Assign a technology-name and phylum. The technology-name selected for the discover is "DIAMOND-CVD-COATING." The corresponding phylum is "SURFACE-TREATMENT."

Assign values to slots. Property values gathered for the various attributes were assigned to the available slots for each class template. The technology-name and phylum were assigned to the appropriate slots for each class.

Assign instance names. Instance names were assigned based on the guidelines presented earlier. The prefix for all instances was "DIAMOND-CVD." The following instance name suffixes were used: "-K" (thermal conductivity), "-C" (specific heat), "-ALPHA" (thermal expansion coefficient), "-ALPHA-DIFF" (thermal diffusivity), "-USE-TEMP" (maximum usage temperature), "-MELT-POINT" (melting point), "-E" (Young's modulus), "-G" (shear modulus), "-NU" (Poisson's ratio), "-RHO" (density), "-R" (hardness), "-HK" (hardness), "-OX-RESIST" (oxidation resistance) and "-LUB-COMPAT" (lubricant compatibility).

Create source file. The instances were encoded using the `definstance` construct. The instances were grouped under the following definstances: "thermal-surface-properties," "mechanical-surface-properties" and "environmental-surface-properties." The source listing, "CVD-Diamond instances.CLP," is located in the Appendix.

Load instances into knowledge base. The instances are loaded into the knowledge base using a batch file. The batch file listing is provided in the Appendix and is examined in more detail in section 7.4.2.

7.2.3.2 Governing equations

Gather attribute information. Equations developed in Chapter 3 were collected and entered into the work sheets summarized in Tables 7-8 and 7-9.

Select appropriate class templates. The class templates, provided in the Appendix, were consulted and the class EMPIRICAL-RELATION was selected.

Assign a technology-name and phylum. Same as previous attribute categories.

Assign values to slots. The technology-name and phylum slots are the same for all members of the DIAMOND-CVD technology class. The c-header slot contains C language function prototypes for each equation instance. These function prototypes will be

Table 7-8. Work sheet for DIAMOND-CVD empirical dimensionless relationships.

Technology Name: DIAMOND-CVD		Phylum: SURFACE-TREATMENT	
Relationship Name	Ref. Eq. No.	c header	Instance Name
THETA-SSO	(3.8)	float ThetaSSO(float PenDepth, float Length, float LayerConductivity, float SubstrateConductivity, float LayerThickness, float EntryLength);	DIAMOND-CVD-THETA-SSO
THETA-OS	(3.9)	float ThetaOS(float PenDepth, float Length, float LayerConductivity, float SubstrateConductivity, float LayerThickness, float EntryLength);	DIAMOND-CVD-THETA-OS
THETA-S	(3.10)	float ThetaS(float PenDepth, float Length, float LayerConductivity, float SubstrateConductivity);	DIAMOND-CVD-THETA-S
THETA-SO	(3.12)	float ThetaSO(float ThetaSSO, float ThetaS);	DIAMOND-CVD-THETA-SO
THETA-O	(3.11)	float ThetaO(float ThetaOS, float ThetaS);	DIAMOND-CVD-THETA-O
THETA-SS	(3.13)	float ThetaSS(float ThetaInterface, float ThetaSO_IFSS, float ThetaO_IFSS);	DIAMOND-CVD-THETA-SS
DELTA	(3.14)	float Delta(float ThetaUncoatedSubstrate, float ThetaSubstrate);	DIAMOND-CVD-DELTA
THETA-SS-P	(3.15)	float ThetaSS_P(float Delta, float ThetaSubstrate);	DIAMOND-CVD-THETA-SS-P
THETA-IF	(3.15)	float ThetaIF(float Delta, float Theta_IF);	DIAMOND-CVD-THETA-IF
THETA-D	(3.15)	float ThetaD(float Delta, float Theta_D);	DIAMOND-CVD-THETA-D

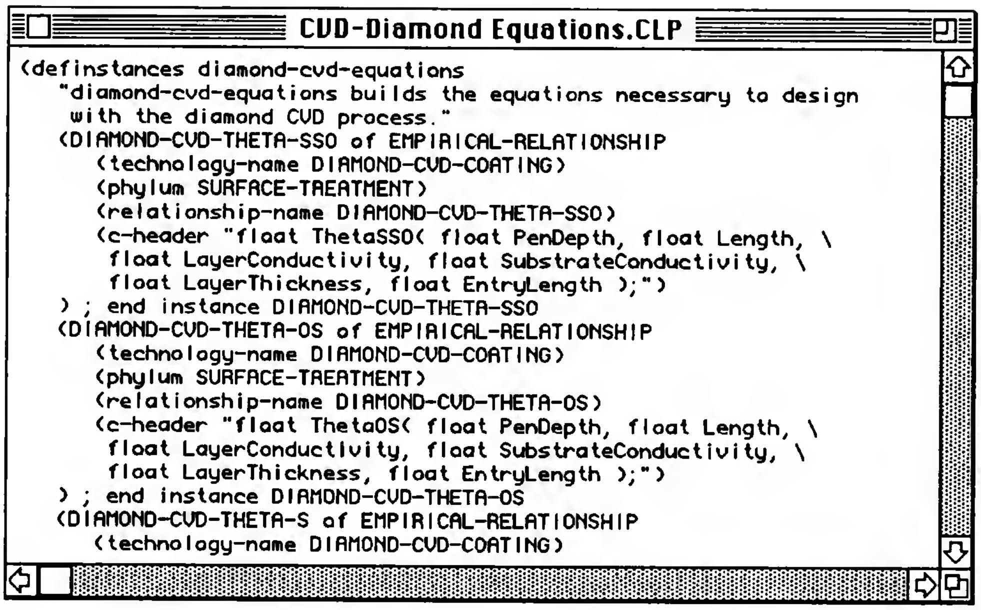
Table 7-9. Work sheet for DIAMOND-CVD empirical dimensional relationships.

Technology Name: DIAMOND-CVD		Phylum: SURFACE-TREATMENT	
Relationship Name	Ref. Eq. No.	c header	Instance Name
TEMPERATURE	(3.16)	float Temperature(float Theta, float HeatIn);	DIAMOND-CVD-T
PEN-DEPTH	(3.6)	float PenetrationDepth(float Conductivity, float Density, float SpecificHeat, float Length, float Velocity);	DIAMOND-CVD-D-PEN
L-ENTRY	(3.7)	float EntryLength(float LayerConductivity, float LayerDensity, float LayerSpecificHeat, float Velocity, float LayerThickness);	DIAMOND-CVD-L-ENTRY
SIGMA	(3.17)	float NormalStress(float YoungsModulus, float ThermalExpansion, float TemperatureRise);	DIAMOND-CVD-SIGMA
TAU	(3.20)	float ShearStress(float LayerNormalStress, float SubstrateNormalStress, float LayerThickness, float LayerWidth);	DIAMOND-CVD-TAU
TEMPERATURE -UNCOATED	(3.16)	float TemperatureS(float ThetaUncoatedSubstrate, float HeatIn);	DIAMOND-CVD-TS
TEMPERATURE -SUBSTRATE	(3.16)	float TemperatureSubstrate(float ThetaSubstrate, float HeatIn);	DIAMOND-CVD-TSS
TEMPERATURE -INTERFACE	(3.16)	float TemperatureInterface(float ThetaInterfaceLayer, float HeatIn);	DIAMOND-CVD-TIF
TEMPERATURE -DIAMOND	(3.16)	float TemperatureDiamond(float ThetaDiamond, float HeatIn);	DIAMOND-CVD-TD

used later in the algorithmic module to model the behavior of the diamond surface coating. It is understood that a library containing all the source code for the prototyped equations is available.

Assign instance names. The instance names are all prefixed with "DIAMOND-CVD." The following suffixes were used for the instance names: "THETA-SSO," "THETA-OS," "THETA-S," "THETA-SO," "THETA-O," "-DELTA," "THETA-SS-P," "THETA-IF," "THETA-D," "-T," "-PEN," "-L-ENTRY," "-SIGMA," "-TAU," "-TS," "-TSS," "-TIF" and "-TD."

Create source file. The source file—"CVD-Diamond Equations.CLP"— is listed in its entirety in the Appendix. The file contains a single `definstances` construct, "diamond-cvd-equations," and is shown in Figure 7-5 while in development.



CVD-Diamond Equations.CLP

```
(definstances diamond-cvd-equations
   "diamond-cvd-equations builds the equations necessary to design
    with the diamond CVD process."
   (DIAMOND-CVD-THETA-SSO of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-THETA-SSO)
      (c-header "float ThetaSSO( float PenDepth, float Length, \
       float LayerConductivity, float SubstrateConductivity, \
       float LayerThickness, float EntryLength );")
   ) ; end instance DIAMOND-CVD-THETA-SSO
   (DIAMOND-CVD-THETA-OS of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-THETA-OS)
      (c-header "float ThetaOS( float PenDepth, float Length, \
       float LayerConductivity, float SubstrateConductivity, \
       float LayerThickness, float EntryLength );")
   ) ; end instance DIAMOND-CVD-THETA-OS
   (DIAMOND-CVD-THETA-S of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
```

Figure 7-5. Source file "CVD-Diamond instances.CLP" under development within the CLIPS environment.

Load instances into knowledge base. The instances are loaded into the knowledge base using a batch file. The batch file listing is provided in the Appendix and is examined in more detail in section 7.4.2.

7.2.4 APPLICATION Instances

Instances for two APPLICATION technology classes are developed in this section. The first class is standard dedendum gear pairs. The relationships for this technology area were developed in Chapter 5. The second class is highly loaded, well lubricated roller bearings. It is shown later that the roller bearing case is not a good qualitative match for the diamond process reported in this study. The roller bearing case provides competing technology APPLICATION attributes for the knowledge base to sort through.

Instances of the two APPLICATION classes are created using different methods. The gear pair physical attributes are created within the knowledge base through the decomposition of their qualitative requirements. The bearings attributes are coded directly into the knowledge base. It is possible to mix and match these methods—using decomposition for some requirements and direct encoding for other requirements.

7.2.4.1 Qualitative requirements for standard dedendum gear pairs

Gather attribute information. Gear pairs in mesh experience high contact stress in the meshing surface and high heat generation because the meshing surfaces are sliding and rolling. Therefore, gears require materials that have good contact stress capability and can handle high, localized heat loads. The qualitative requirements for gears are summarized in Table 7-10.

Select appropriate class templates. The QUALITATIVE-REQTS class as defined in the Appendix was used as a template for the gear pair qualitative requirements.

Assign a technology-name and phylum. "GEARS" is the assigned technology name. The phylum is classified as "MECHANICAL-POWER-TRANSMISSION."

Assign values to slots. The slot values for the "CONTACT-STRESS" and "SLIDE-TO-ROLL-RATIO" criteria were determined to be "HIGH."

Table 7-10. Qualitative requirements for GEARS.

Technology Name: GEARS		Phylum: MECHANICAL-POWER-TRANSMISSION	
Criterion Name	Criterion Category	Qualitative Value	Instance Name
CONTACT-STRESS	SURFACE-CONTACT	HIGH	GEAR-CONTACT-STRESS
SLIDE-TO-ROLL-RATIO	SURFACE-CONTACT	HIGH	GEAR-SLIDE-ROLL

Assign instance names. Instance names assigned to the criteria were "GEAR-CONTACT-STRESS" and "GEAR-SLIDE-ROLL," respectively.

Create source file. The `definstances` construct, "gear-application," was encoded in the file "Gear Instances.CLP," which is listed in the Appendix.

Load instances into knowledge base. As in previous cases, the instances are loaded into the knowledge base using a batch file. The batch file listing is provided in the Appendix and is examined in more detail in section 7.4.2.

7.2.4.2 Governing equations for standard dedendum gear pairs

Gather attribute information. C language declarations for the gear design parameters were collected and are summarized in Table 7-11. The empirical relationships for stan-

Table 7-11. Work sheet for GEARS design parameters.

Technology Name: GEARS	Phylum: MECHANICAL-POWER-TRANSMISSION	
Parameter Name	c declaration	Instance Name
CENTER-DISTANCE	float CenterDistance;	GEAR-CD
N-PINION-TEETH	int NPinion;	GEAR-NP
GEAR-RATIO	int GearRatio;	GEAR-MG
PINION-SPEED	float omega;	GEAR-OMEGA
INPUT-TORQUE	float TorqueIn;	GEAR-TORQUE

dard dedendum gear pairs are given in Tables 7-12 and 7-13. Constants for use in controlling the bounds of the gear design space are provided in Table 7-14.

Select appropriate class templates. The following class templates were used as a guide for building gear pair governing equation instances: "EMPIRICAL-RELATIONSHIP," "DESIGN-PARAMETER" and "CONSTANT."

Assign a technology-name and phylum. For consistency, "GEARS" was used as the technology-name and "MECHANICAL-POWER-TRANSMISSION" as the phylum.

Assign values to slots. Values assigned to the instance slots were taken from the previously defined work sheets.

Assign instance names. The instance names for the three classes used in defining the gear equation objects are given in Tables 7-12, 7-13 and 7-14.

Create source file. The instances were created using the following `definstances` constructs: "gear-design-parameters," "gear-constants" and "gear-equations." The source listing, "Gear Equations.CLP" is included in the Appendix.

Load instances into knowledge base. As in previous cases, the instances are loaded into the knowledge base using a batch file. The batch file listing is provided in the Appendix and is examined in more detail in section 7.4.2.

7.2.4.3 Qualitative requirements for highly loaded, lubricated roller bearings

Highly loaded roller bearings operate in an almost pure rolling mode with an insignificant amount of sliding. This absence of sliding, coupled with the benefits of ample lubrication, eliminates most thermal deterioration effects. Material properties in this case do not have to be extreme, as in the case of the gear systems under investigation.

Gather attribute information. Table 7-15 summarizes required thermal properties of highly loaded roller bearings. Required mechanical properties are provided in Table 7-16 and Table 7-17 sums up required environmental properties.

Select appropriate class templates. Class templates selected include the following: THERMAL-CONDUCTIVITY, SPECIFIC-HEAT, THERMAL-EXPANSION, USAGE-

Table 7-12. Work sheet for GEARS empirical relationships.

Technology Name: GEARS		Phylum: MECHANICAL-POWER-TRANSMISSION	
Relationship Name	Ref. Eq. No.	c header	Instance Name
Q-PINION	(5.1)	float GearQPinion(float KrhoC, float DeltaTPinion, float aWidth, float VelocityPinion, float psiP);	GEAR-Q-PINION
ROLL-VELOCITY	(5.2)	float GearRollVelocity(float Vpinion, float Vgear);	GEAR-V-ROLL
SLIDE-VELOCITY	(5.3)	float GearSlideVelocity(float Vpinion, float Vgear);	GEAR-V-SLIDE
PINION-VELOCITY	(5.4)	float GearPinionVelocity(float omegaPinion, float CenterDistance, float X, float phi);	GEAR-V-PINION
GEAR-VELOCITY	(5.5)	float GearGearVelocity(int GearRatio, float omegaPinion, float CenterDistance, float X, float phi);	GEAR-V-GEAR
CONTACT-WIDTH	(5.6)	float GearContactWidth(float TangentLoad, float EEffect, float RadiusEffect, float phi);	GEAR-A-WIDTH
RADIUS-EFFECTIVE	(5.7)	float GearEffectiveRadius(float CenterDistance, float X, float phi);	GEAR-R-EFF
YOUNGS-MODULUS-EFF	(5.8)	float GearEffectiveModulus(float EGear, float EPinion);	GEAR-E-EFF

Table 7-13. Work sheet for GEARS empirical relationships, and temperature and stress prediction.

Technology Name: GEARS		Phylum: MECHANICAL-POWER-TRANSMISSION	
Relationship Name	Ref. Eq. No.	c header	Instance Name
PINION-PITCH-DIA	(5.9)	float GearPinionPitchDia(float CenterDistance, int GearRatio);	GEAR-PINION-PD
T-PINION-START	(5.10)	float GearTPinionStart(float X, int GearRatio, float phi, float psiP);	GEAR-TPS
X-START	(5.11)	float GearXStart(int GearRatio, int NPinion, float phi);	GEAR-XSTART
T-PINION	(5.12)	float GearTPinion(float TPS, float f, float EEquiv, float omegaPinion, float CenterDistance, float KrhoC);	GEAR-TP
BENDING-STRESS	(5.13)	float GearBendingStress(float TangentLoad, int NPinion, float CenterDistance, int GearRatio);	GEAR-SIGMA-B
CONTACT-STRESS	(5.14)	float GearContactStress(float TangentLoad, float EEquiv, float RadiusEquiv, float phi);	GEAR-SIGMA-H

Table 7-14. Work sheet for GEARS constants.

Technology Name: GEARS		Phylum: MECHANICAL-POWER-TRANS-MISSION
Parameter Name	c define	Instance Name
GEAR-TEMP-PINION-MAX RISE	#DEFINE GEAR-TEMP-PIN-ION-MAX RISE *	GEAR-TP-MAX
GEAR-TEMP-GEAR-MAX RISE	#DEFINE GEAR-TEMP-GEAR-MAX RISE *	GEAR-T-G-MAX
GEAR-MAX-BENDING-STRESS	#DEFINE GEAR-MAX-BEND-ING-STRESS	GEAR-SIGMA-B-MAX
GEAR-MAX-CONTACT-STRESS	#DEFINE GEAR-MAX-CON-TACT-STRESS	GEAR-SIGMA-H-MAX

Table 7-15. Work sheet for HI-LOAD- ROLLER-BRG thermal-physical properties.

Technology Name: HI-LOAD-ROLLER-BRG			Phylum: ANTI-FRICTION-BRG		
Property Name	Quantitative		Qualitative		Instance Name
	Value	Range	Value	Range	
thermal con-ductivity			medium		HI-LOAD-ROLLER-BRG-K
specific heat			medium		HI-LOAD-ROLLER-BRG-C
thermal expansion coefficient			medium		HI-LOAD-ROLLER-BRG-ALPHA
maximum usage temper-ature			very high		HI-LOAD-ROLLER-BRG-USE-TEMP

TEMPERATURE, YOUNGS-MODULUS, SHEAR-MODULUS, POISSON-RATIO, DENSITY, ROUGHNESS, OXIDATION-RESISTANCE and LUB-COMPATIBILITY.

Assign a technology-name and phylum. The assigned technology-name is "HI-LOAD-ROLLER-BRG." The corresponding phylum is "ANTI-FRICTION-BRG."

Table 7-16. Work sheet for HI-LOAD- ROLLER-BRG mechanical-physical properties.

Technology Name: HI-LOAD-ROLLER-BRG			Phylum: ANTI-FRICTION-BRG		
Property Name	Quantitative		Qualitative		Instance Name
	Value	Range	Value	Range	
Young's modulus			medium high		HI-LOAD-ROLLER-BRG-E
shear modulus			very high		HI-LOAD-ROLLER-BRG-G
Poisson's ratio			medium		HI-LOAD-ROLLER-BRG-NU
density			medium low		HI-LOAD-ROLLER-BRG-RHO
surface roughness			low		HI-LOAD-ROLLER-BRG-R
hardness (Knoop)			medium high		HI-LOAD-ROLLER-BRG-HK

Assign values to slots. Slot values are summarized in Tables 7-15, 7-16 and 7-17.

Assign instance names. The prefix for all instances was "HI-LOAD-ROLLER-BRG." The following instance name suffixes were used: "-K" (thermal conductivity), "-C" (specific heat), "-ALPHA" (thermal expansion coefficient), "-USE-TEMP" (maximum usage temperature), "-E" (Young's modulus), "-G" (shear modulus), "-NU" (Poisson's ratio), "-RHO" (density), "-R" (surface roughness), "-OX-RESIST" (oxidation resistance) and "-LUB-COMPAT" (lubricant compatibility).

Create source file. The source file, "Hi-Load-Brg Instances.CLP," is listed in the Appendix.

Table 7-17. Work sheet for HI-LOAD-ROLLER-BRG environmental-physical properties.

<table>
<tr><td colspan="3">Technology Name: HI-LOAD-ROLLER-BRG</td><td colspan="3">Phylum: ANTI-FRICTION-BRG</td></tr>
<tr><td rowspan="2">Property Name</td><td colspan="2">Quantitative</td><td colspan="2">Qualitative</td><td rowspan="2">Instance Name</td></tr>
<tr><td>Value</td><td>Range</td><td>Value</td><td>Range</td></tr>
<tr><td>oxidation resistance</td><td></td><td></td><td>medium</td><td></td><td>HI-LOAD-ROLLER-BRG-OX-RESIST</td></tr>
<tr><td>Property Name</td><td colspan="4">List of Items</td><td>Instance Name</td></tr>
<tr><td>compatible lubricants</td><td colspan="4">oil</td><td>HI-LOAD-ROLLER-BRG-LUB-COMPAT</td></tr>
</table>

Load instances into knowledge base. As in previous cases, the instances are loaded into the knowledge base using a batch file. The batch file listing is provided in the Appendix and is examined in more detail in section 7.4.2.

7.3 Building Rules

Part of the power knowledge base management systems have over traditional relational database management systems stems from the ability to incorporate heuristic information along with the data. The heuristics, captured in the form of rules, increase the level of semantic sophistication required to model complex system behavior. There are two important types of rules used in the developed system. The first rule type decomposes qualitative requirements of APPLICATION instances into matchable attribute instances. The second rule type performs the matching between qualitative attributes of DISCOVERY and APPLICATION technology instances.

7.3.1 Decomposing Qualitative Requirements

One rule to handle decomposition of all QUALITATIVE-REQTS instance types is impractical. Instead, as new QUALITATIVE-REQTS instances are created, new decom-

position rules should be devised. This methodology requires the knowledge base developer to be very familiar with the physics of the APPLICATION attribute domain. As more and more decomposition rules are developed, the system will be able to adapt to more and more general QUALITATIVE-REQTS instances, thus reducing the burden of introducing new rules for every new APPLICATION.

7.3.1.1 Contact stress

DECOMPOSE-CONTACT-STRESS is a sample rule that transforms high contact stress and high slide-to-roll ratio qualitative attribute requirements into individual attribute needs. The rule searches for instances of the same APPLICATION with CONTACT-STRESS and SLIDE-TO-ROLL-RATIO criteria defined. If both instances have high qualitative values, then a new set of attribute instances are created for the APPLICATION. For this particular rule, instances of the THERMAL-CONDUCTIVITY, SPECIFIC-HEAT and DENSITY are created in the action side of the `defrule` DECOMPOSE-CONTACT-STRESS. Pseudo-code describing the rule structure is shown in Figure 7-6.

```
Rule: DECOMPOSE-CONTACT-STRESS
for a given APPLICATION qualitative-reqt
   if CONTACT-STRESS is HIGH or MEDIUM-HIGH or VERY-HIGH
   and
   if SLIDE-TO-ROLL-RATIO is HIGH or MEDIUM-HIGH or VERY-HIGH
   then create the following instances for the given
      APPLICATION:
         THERMAL-CONDUCTIVITY
         SPECIFIC-HEAT
         DENSITY
```

Figure 7-6. Pseudo-code description of the DECOMPOSE-CONTACT-STRESS rule.

7.3.1.2 Pure rolling

For the case of high contact stress and pure rolling (slide-roll-ratio is very small), the thermal properties are of less consequence. In this regime, the high temperatures due to friction are eliminated. More important for this application are lubricant compatibility and high hardness. It would not be difficult to construct a rule for this purpose. This rule would be applicable to the HI-LOAD-ROLLER-BRG technology, but since instances of this class were coded directly into the knowledge base, this rule was not pursued further.

7.3.2 Qualitative Matching

The qualitative matching process works by examining instances of the DISCOVERY and APPLICATION technology classes for corresponding species slot values. Match table entries are created whenever a match is found. Each attribute match is rated for "goodness of fit" and the overall DISCOVERY-APPLICATION match is scored using a simple methodology. Care must be taken in relying too heavily on the numerical score of a technology match.

7.3.2.1 Heuristics for matching

The basis behind the matching process is to compare requirements of the applications with capabilities of the discovery. The match heuristic first looks at instances of DISCOVERY properties that also have the property in an APPLICATION instance. Once the match is made, then the "goodness of fit" is evaluated for the match. A strong fit is considered when the qualitative properties are within one level of each other. For instance, if the discovery's thermal conductivity is `VERY-HIGH` and the application requires a `HIGH` thermal conductivity, then it would be considered a strong match for that criteria. The more strong matches that occur for a discovery-application pair, then the higher the ranking.

7.3.2.2 Overall match rating scheme

A simple rating scheme for evaluating the overall DISCOVERY-APPLICATION "goodness of fit" was implemented in the prototype matching knowledge base system. A

score that relates the total number of STRONG matches to the number of total matches is used to pass a preliminary rating onto the knowledge base user. For instance, if a DISCOVERY-APPLICATION pair has 10 matching attribute instances, of which 6 are STRONG, then the score is 0.60. In this scheme, the higher the score, the better the overall fit.

7.3.2.3 Match rating scheme caveats

One must take care in relying too heavily on this scoring scheme. Suppose that a DISCOVERY-APPLICATION pair scores a 0.90. What happens if the attribute that didn't match is critical—say the surface treatment DISCOVERY dissolves in the APPLICATION technology's lubricant? This is obviously not an ideal match. The matching results must therefore be scrutinized. Engineering judgement cannot be left out of the process.

The mismatch on critical attributes can be used to direct further research efforts. A big performance payoff will result from using the diamond coating process on steel. Diamond's low thermal expansion relative to steel and steel's ability to dissolve carbon from the diamond interface make it currently very difficult to apply. Much of the work in this reported study was directed at the development of a buffer layer scheme to show the promise of the new diamond technology—concentrating on "what can be" rather than "what is."

7.4 Sample Execution of the Qualitative Matching Engine

7.4.1 Overview of the Process

Once all the instances of the technology classes have been defined, the user is ready to begin the qualitative matching process. Executing the system is as simple as launching the CLIPS system, loading the constructs into the system and running it. When the matching process is complete, the user has a variety of options for viewing the associated match statistics and writing pertinent equations to external files for analytical post-processing. The process flow is diagrammed in Figure 7-7.

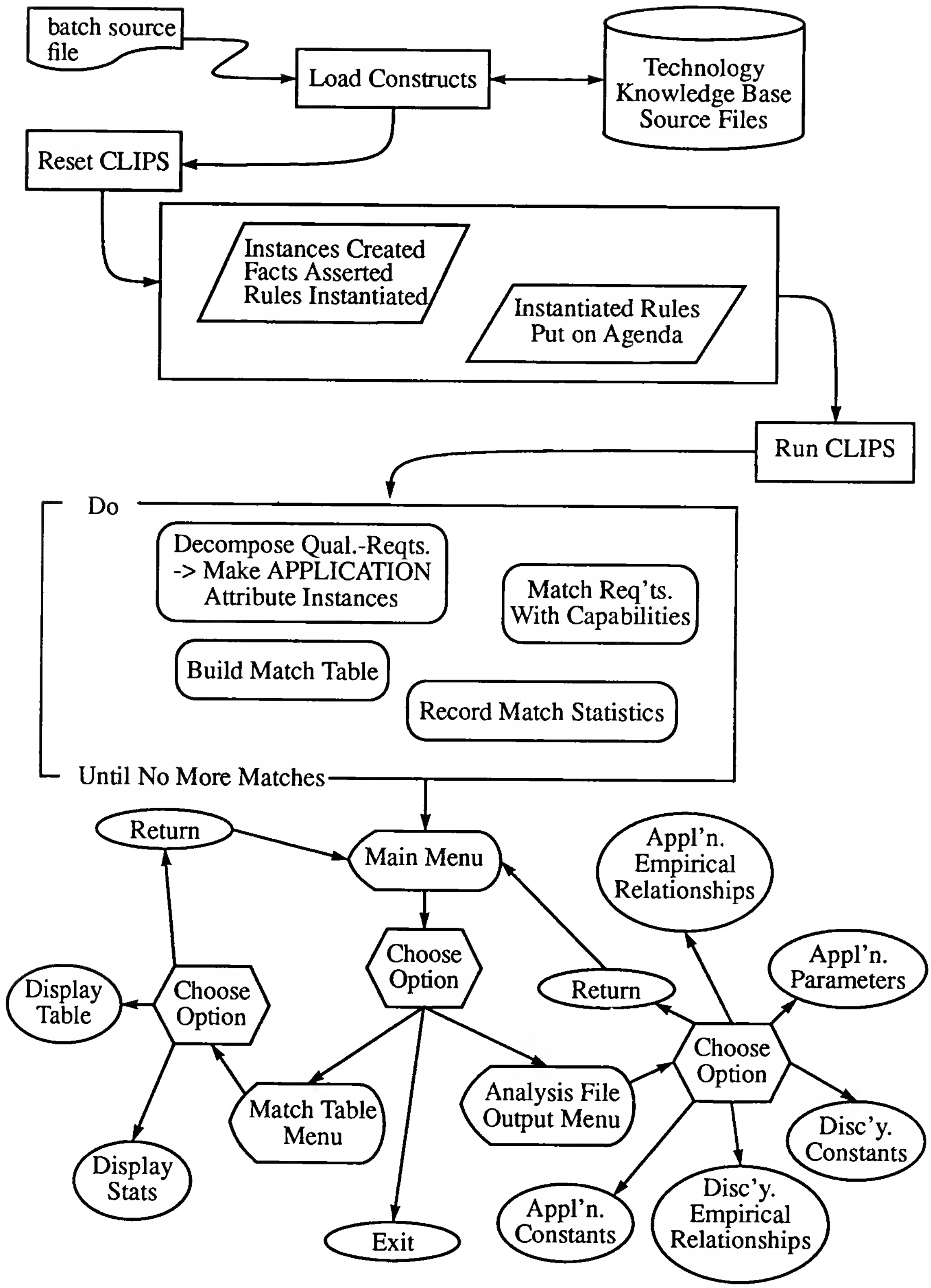


Figure 7-7. Loading and execution of qualitative matching process knowledge base system within the CLIPS environment.

7.4.2 Loading the Knowledge Base Constructs

The constructs defining the knowledge base are divided into many files to logically separate the many entity types by function. As mentioned in the previous chapter, it is important to load the constructs in a particular order so as to prevent errors. Since there are so many files to deal with, it is best to automate the loading process via a CLIPS batch file.

7.4.2.1 Batch file organization

The batch file used in the prototype system, "Technology KBS.bat," is organized as shown in Figure 7-8. A complete listing of "Technology KBS.bat," is provided in the Appendix.

```
(clear)
; define templates for match table tracking
(load "Match Templates.CLP")
; define class hierarchy for the new technology knowledge base
     load defclasses here. . .
; define instances of the technology knowledge base attributes
     load definstances here. . .
; define generic methods
(load "Generic Methods.CLP")
; define functions
     load deffunctions here. . .
; define rules
     load defrules here. . .
; make sure we can always find the help file
(help-path "Keith's PPC:Development:CLIPS:CLIPS
6.0:Executables:clips.hlp")
(reset)
```

Figure 7-8. Skeleton of the CLIPS batch file "Technology KBS.bat."

7.4.2.2 Loading the batch file

The batch file can be loaded into the CLIPS system in several ways. If the CLIPS system is already running, then the "Load Batch..." command can be selected from the

"FILE" menu. A dialog window opens and the user selects the appropriate batch file. An example of a file selection dialog window as implemented on the Macintosh system is shown in Figure 7-9. An alternate method is to type `(load "Technology KBS.-`

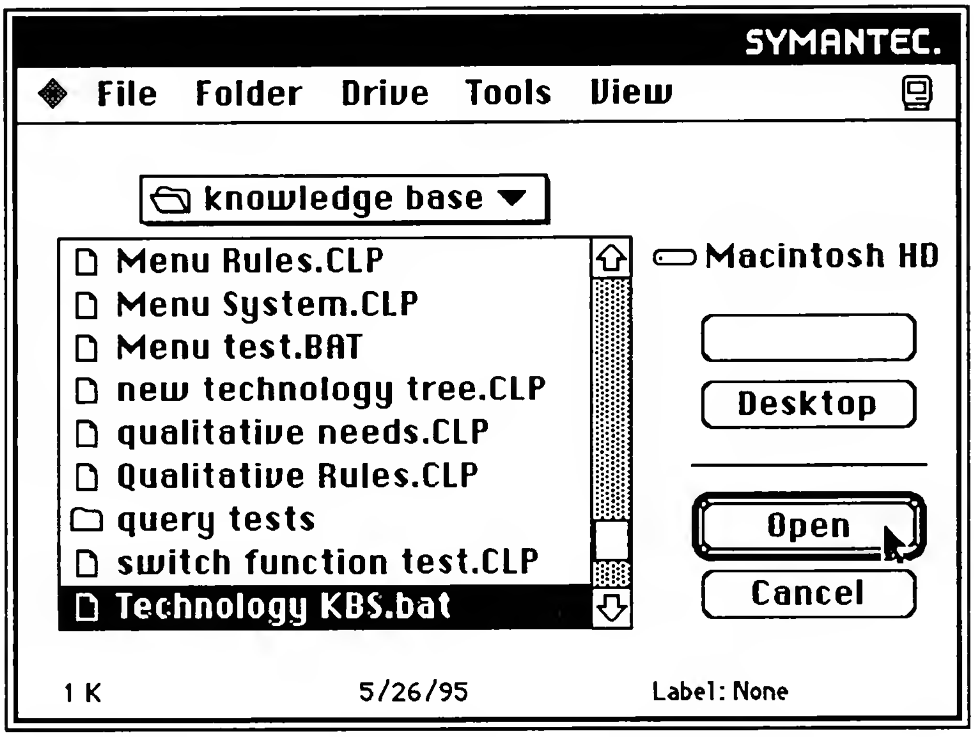


Figure 7-9. Opening the batch file from within the CLIPS environment.

`bat")` at the CLIPS prompt.

If the CLIPS system is not currently running, then the icon for the batch file may be opened directly from the computer's desktop. On the Macintosh, this is accomplished by double-clicking on the icon with the mouse. Figure 7-10 shows this operation in progress. To launch the CLIPS system without starting up the batch file, double-click on the CLIPS icon with the mouse as shown in Figure 7-10.

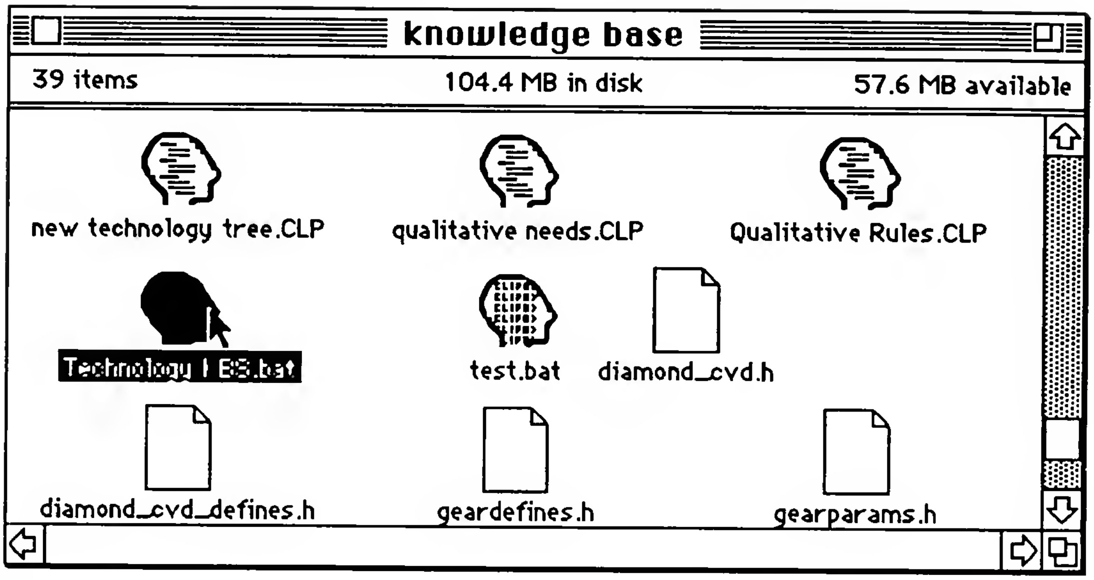


Figure 7-10. Launching the batch file directly from the Desktop.

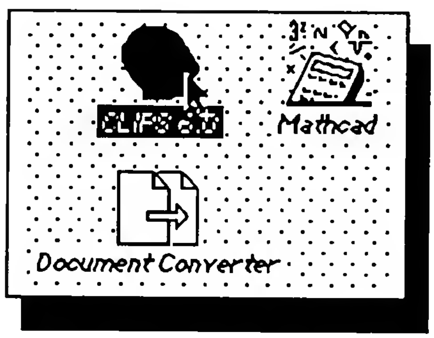


Figure 7-11. Launching CLIPS directly from the Desktop.

7.4.3 Executing the system

Once the batch file is loaded by the user, the only operation necessary to start the matching process is to run the system. From the CLIPS prompt, the user simply types "`(run)`," and the process is underway. Naturally, there is an alternative method. From the CLIPS "EXECUTION" menu, select the "Run" option. After a short period of time, the main knowledge base application menu appears as shown in Figure 7-12.

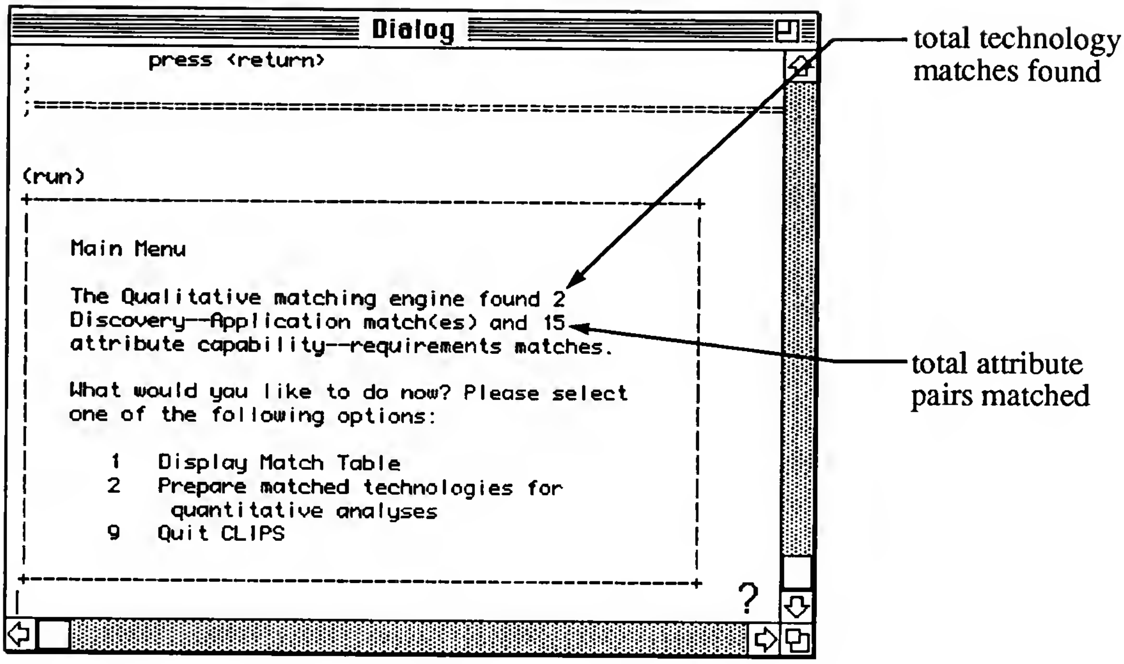


Figure 7-12. Main menu display.

Match table menu. Selecting option 1 from the main menu sends the program control to the match table menu. This menu allows the user to view and save to file match table information and match statistics. Match table entries review each attribute pair matched during the execution of the system. The match table menu is shown in Figure 7-13. Sample match table entries are illustrated in Figure 7-14. A complete listing of the matches found for the technology instances created in this chapter can be found in the Appendix.

Match statistics are summarized by selecting option 4 from the match table menu. Match statistics for the HI-LOAD-ROLLER-BRG/DIAMOND-CVD-COATING match are shown in Figure 7-15. Notice that the overall match score for this combination is 0.33. The match statistics for the GEARS/DIAMOND-CVD-COATING match are shown in Figure 7-16. Notice that the overall match score for this combination is 1.00.

Output for analysis menu. Once the match statistics have been reviewed, selecting option 9 runs control to the main menu. From here, selecting option 2 opens the output for analysis menu as shown in Figure 7-17. This menu provides the user with the capability to

```
Dialog
+---------------------------------------------------------+
|                                                         |
|   Display Match Table Menu                              |
|                                                         |
|   What would you like to do now? Please select          |
|   one of the following options:                         |
|                                                         |
|       1   Display Match Table                           |
|       2   Save Match Table to an output file            |
|                                                         |
|       4   Display Match Statistics                      |
|       5   Save Match Statistics to an output file       |
|                                                         |
|       9   Return to Main Menu                           |
|                                                         |
+---------------------------------------------------------+
|                                                       ?
```

Figure 7-13. Match table menu display.

```
Dialog
|                              Match Table                              |
| Criterion Matched:                  SPECIFIC-HEAT                     |
| Match Goodness:                            STRONG                     |
| Match Instance Identifier:                  gen29                     |
| Match Identifier:                           gen30                     |
+-----------------------------------------------------------------------+
| Category                  Discovery                 Application       |
+-----------------------------------------------------------------------+
| Name:               DIAMOND-CVD-COATING                      GEARS    |
| Qualitative Value:               MEDIUM                     MEDIUM    |
| Attribute Address:          DIAMOND-CVD-C                  GEARS-C    |
+-----------------------------------------------------------------------+


+-----------------------------------------------------------------------+
|                              Match Table                              |
| Criterion Matched:           THERMAL-CONDUCTIVITY                     |
| Match Goodness:                            STRONG                     |
| Match Instance Identifier:                  gen31                     |
| Match Identifier:                           gen32                     |
+-----------------------------------------------------------------------+
| Category                  Discovery                 Application       |
+-----------------------------------------------------------------------+
| Name:               DIAMOND-CVD-COATING                      GEARS    |
| Qualitative Value:            VERY-HIGH                       HIGH    |
| Attribute Address:          DIAMOND-CVD-K                  GEARS-K    |
+-----------------------------------------------------------------------+
```

Figure 7-14. Match table display showing the evaluation of two attribute matches.

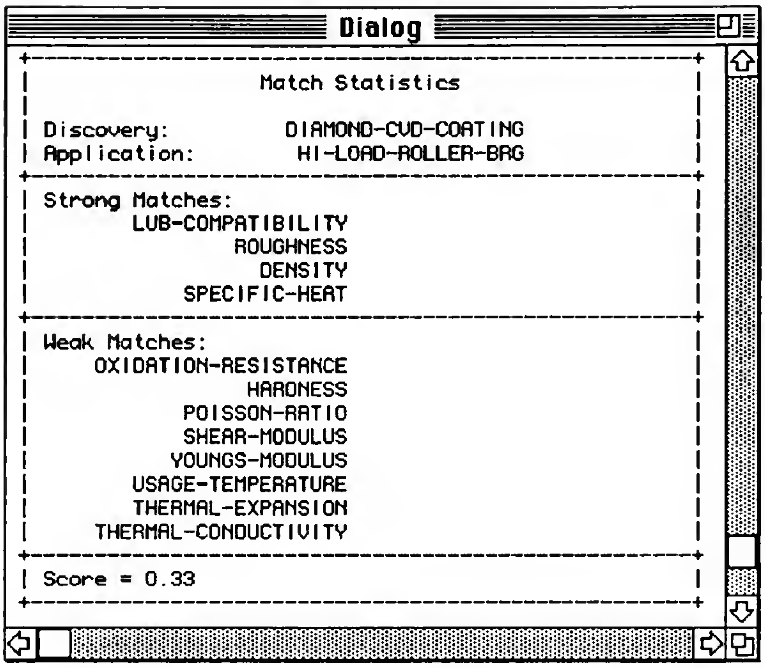


Figure 7-15. Match statistics including the overall score for the DIAMOND-CVD-COATING/HI-LOAD-ROLLER-BRG match.

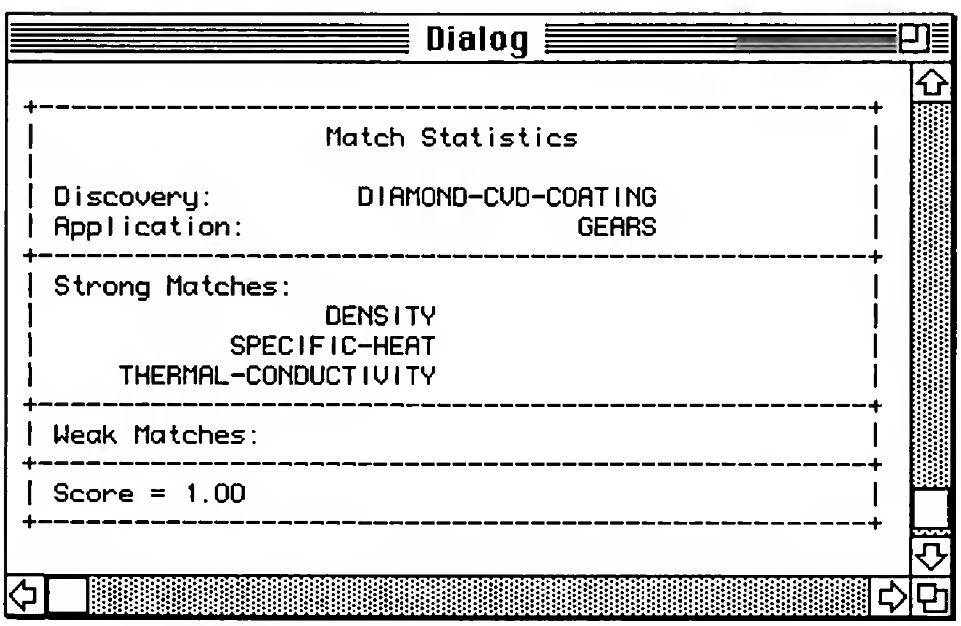


Figure 7-16. Match statistics including overall score for the DIAMOND-CVD-COATING/GEARS match.

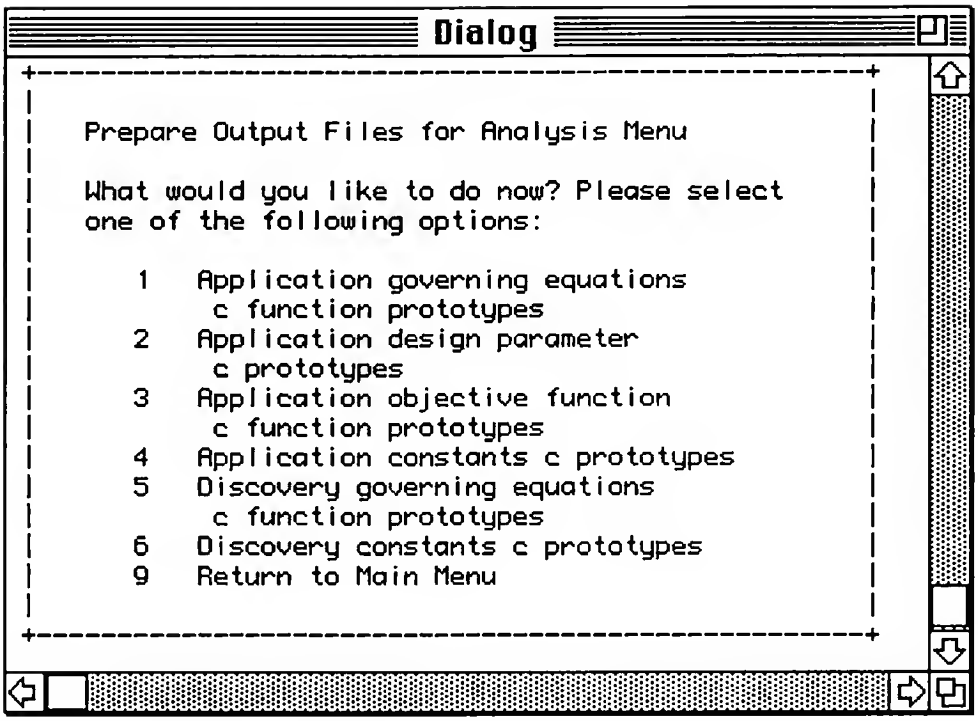


Figure 7-17. Menu display for output of analysis equations.

export C language header files for use later in the analysis module. Options 1 through 6 prompt first for an output file and then proceed to export the requested information. Option 1 produces the function prototypes for the application governing equations. Figure 7-18 shows the resultant file generated from option 1. Design parameters and constants are shown in Figure 7-19.

Governing equations for the DIAMOND-CVD-COATING discovery are illustrated in Figure 7-20. A complete listing of the files generated from this menu is available in the Appendix.

<u>Leaving the menu system</u>. Selecting option 9 returns control to the main menu. Selecting 9 again exits the user from the menu system. At this time, the knowledge base is still fully active and the user can run queries and review the system's contents interactively.

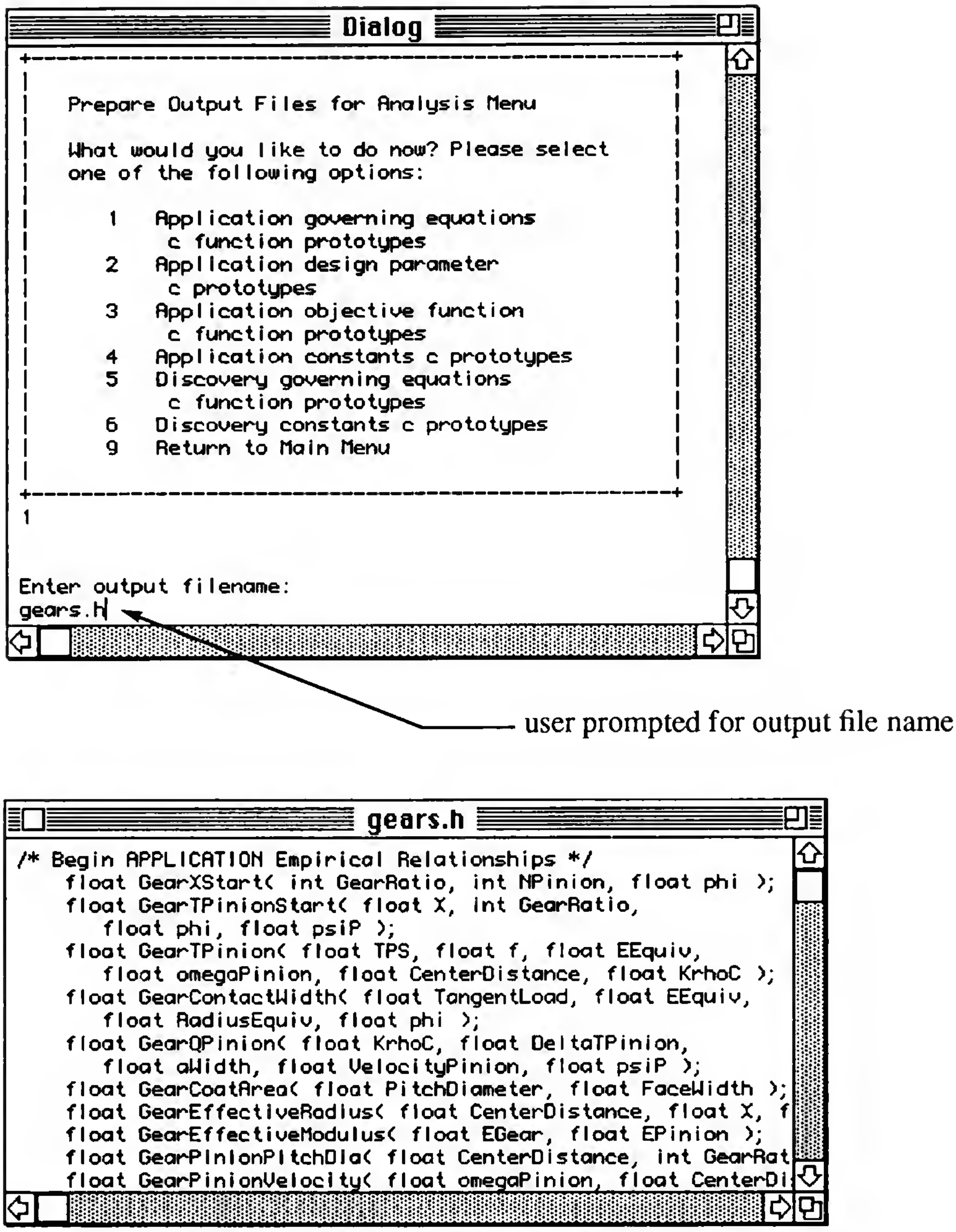


Figure 7-18. Output of governing equations for GEARS in the source file “gears.h.”

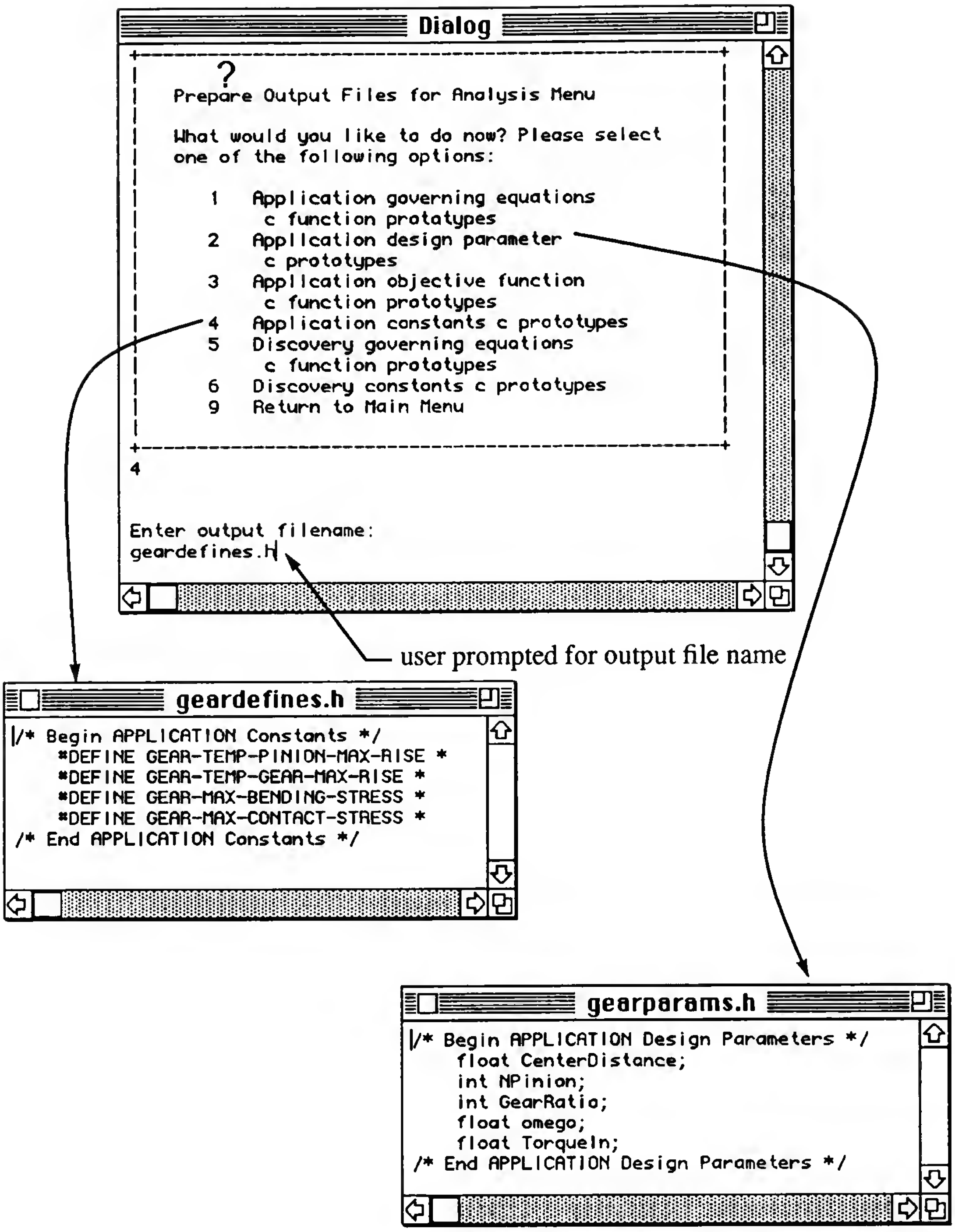


Figure 7-19. Output of design parameters and constants for GEARS in the source files "gearparams.h" and "geardefines.h," respectively.

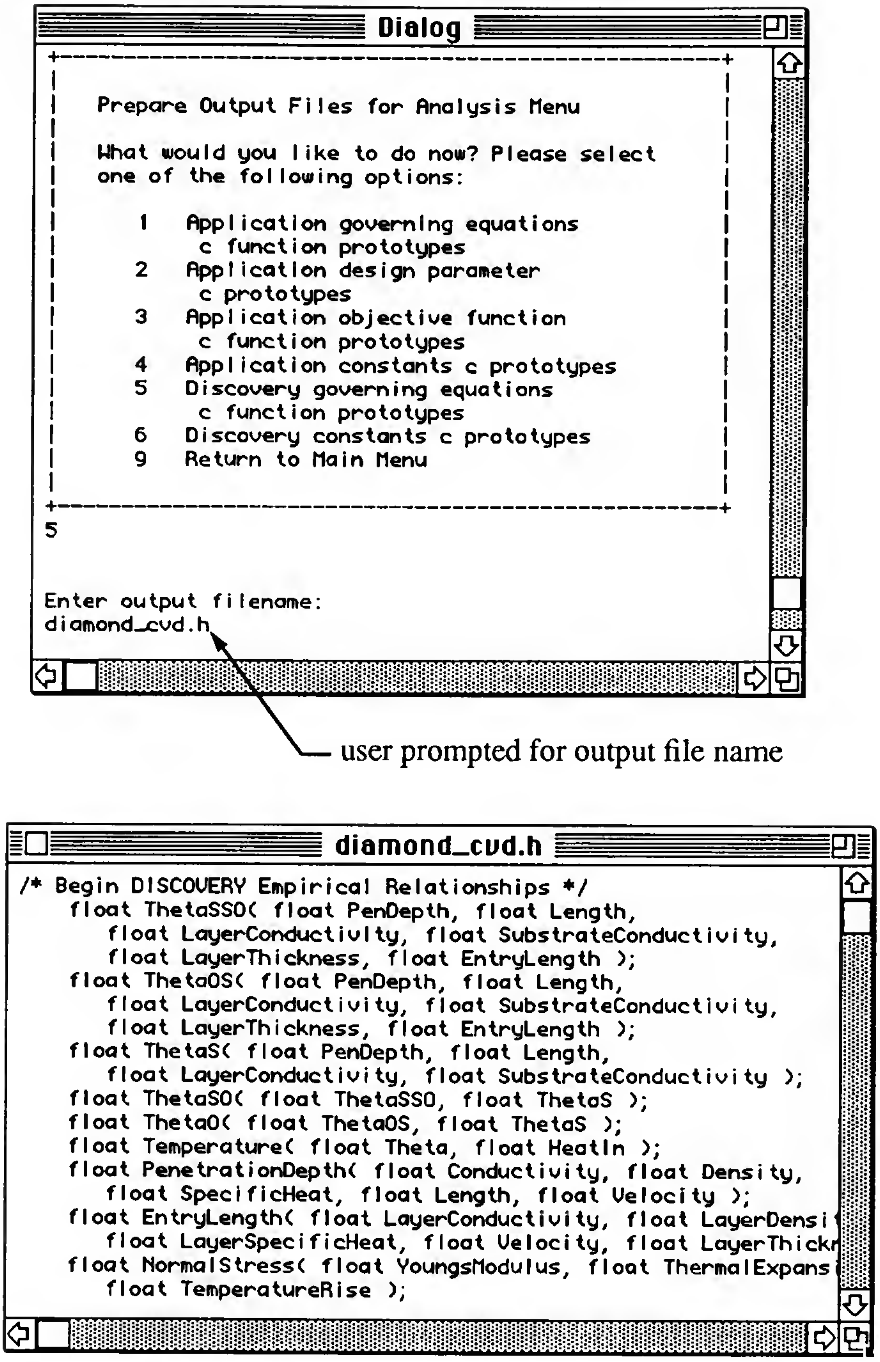


Figure 7-20. Output of governing equations for CVD-DIAMOND-COATING in the source file "diamond_cvd.h."

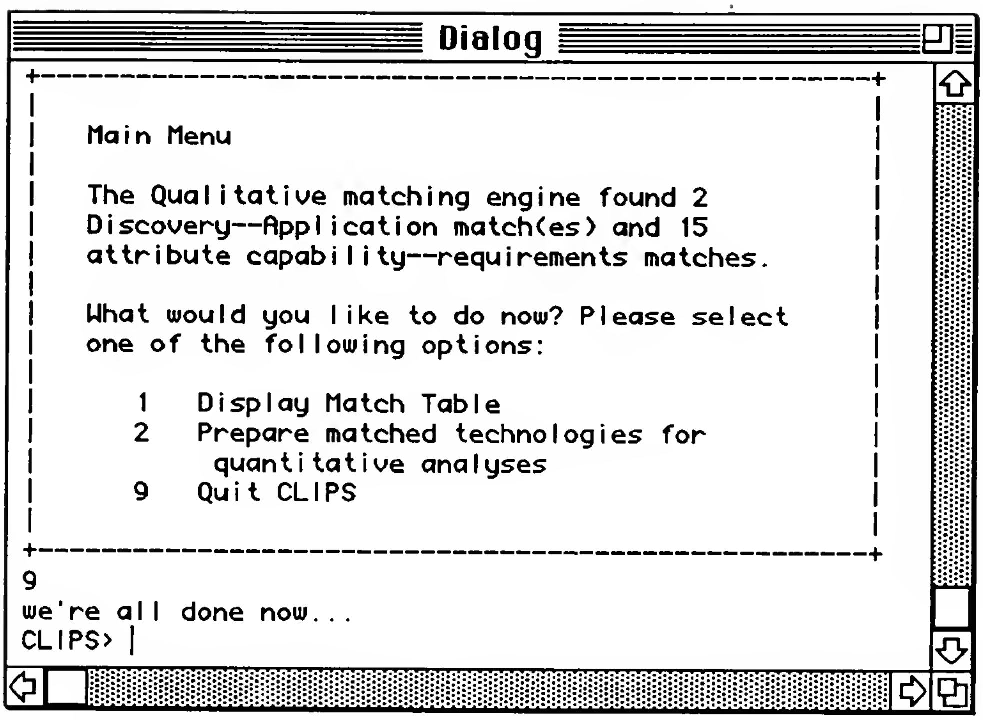


Figure 7-21. Exiting from the menu system.

CHAPTER 8
THE QUANTITATIVE PROCESS

The previous chapter described in detail how to add emerging technological discoveries and potential applications to the qualitative matching knowledge base management system. Two APPLICATION matches—gears (a good match) and highly loaded roller bearings (a bad match)—were found for the discovered diamond process. This chapter develops the quantitative procedure used to evaluate qualitative matches. The methodology involves an algorithmic design simulation and cost/benefit analysis. Results of the cost/benefit analyses are used to rank applications of the new technology. The example from the previous chapter is continued to demonstrate the process.

8.1 Elements of the Quantitative Process

The quantitative matching process is composed of two modules. The first module performs a design simulation to meet performance criteria provided by the user. A second module post-processes the simulation results to produce cost/benefit analyses. After all qualitative matches are evaluated, it is possible to rank the applications. A flow chart illustrating the process is shown in Figure 8-1.

8.1.1 Algorithmic Module

The algorithmic module performs design simulations—based on user-specified criteria—for each qualitative match identified. The module requires the collection of pertinent functional relationships, design parameters and properties for the technology match. Once these entities are gathered, an objective—such as maximum cost/benefit—is formulated and the design constraints are specified. The relationships are then organized into a simulation where cost and performance characteristics can be determined.

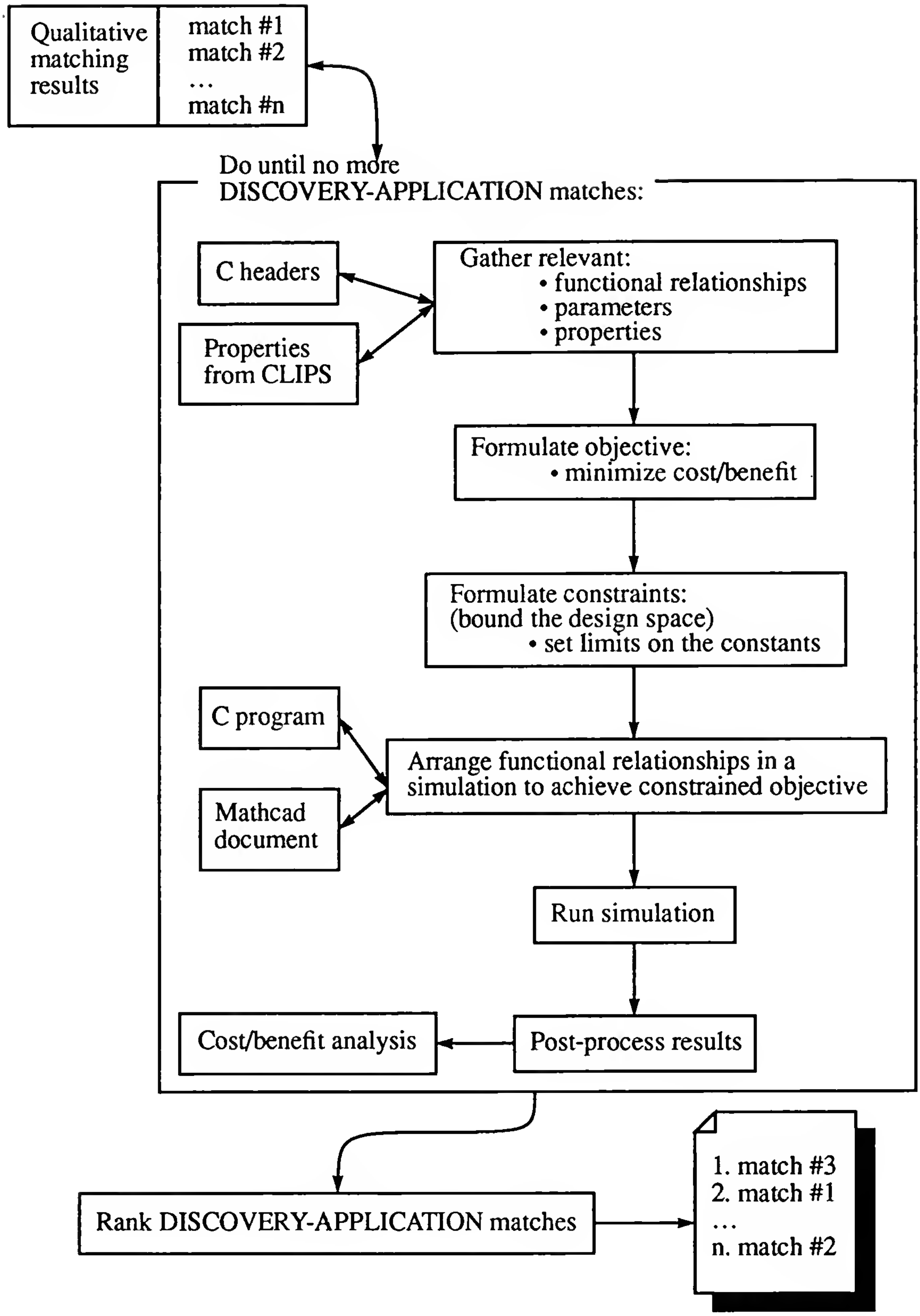


Figure 8-1. The quantitative/algorithmic analysis and ranking process.

8.1.1.1 Algorithmic methodology

Gather functions, parameters and properties. Functional relationships, design parameters and properties may be fed directly to the algorithmic module from the qualitative matching engine or collected by the designer for implementation. These relations should describe the physics of the process and the empirical equations used to design applications of the process.

Formulate the objective. The objective for the algorithmic process is to generate cost and performance characteristics for a technology match. Life improvement in hours is a typical performance characteristic. Others might include decreased response time, increased process yield and increased bandwidth.

Bound the design space. It is critical to constrain the design region for the evaluation of application performance characteristics. For the case of gear pairs, the design space should be limited by maximal contact and thermal stress values and maximum temperature rise. Decision variables should be identified and design parameters set. For instance, gear ratio and center distance for gears may be fixed, while number of teeth in the pinion is allowed to vary.

Build the simulation. The simulation is built using the information gathered above. The procedure can be developed in an interactive fashion using off-the-shelf mathematical modeling tools or through a custom implementation in a computer language of choice.

Run the simulation. The simulation is run to generate the performance characteristics and cost information. This stage includes graphical representation of results so that the design user can quickly spot anomalous results. The performance and cost characteristics determined at this stage are post-processed by a cost/benefit module in preparation for final ranking.

8.1.1.2 Methodology implementation

Long-term implementation. Ultimately it would be beneficial to have a library of simulations ready made for automated integration within the algorithmic system. The qualita-

tive implementation includes a utility to export C language function prototypes and declarations. The burden is on the user to arrange these components within a C program to run a simulation. This process could be made less cumbersome with some clever programming and a good software scripting tool.

Prototype system. Near-term implementation of the algorithmic module was handled by a commercial engineering spreadsheet-like tool. Mathcad—a product of Mathsoft, Inc.—was used to generate all of the simulations used in prototyping the algorithmic system. Mathcad has a document-driven, window interface that allows the user to build equations in standard mathematical notation. Text and imported graphics may be included in a Mathcad document. Equations can be solved numerically or symbolically in Mathcad and results can be plotted in 2D or 3D. Figure 8-2 shows a sample Mathcad document. Note

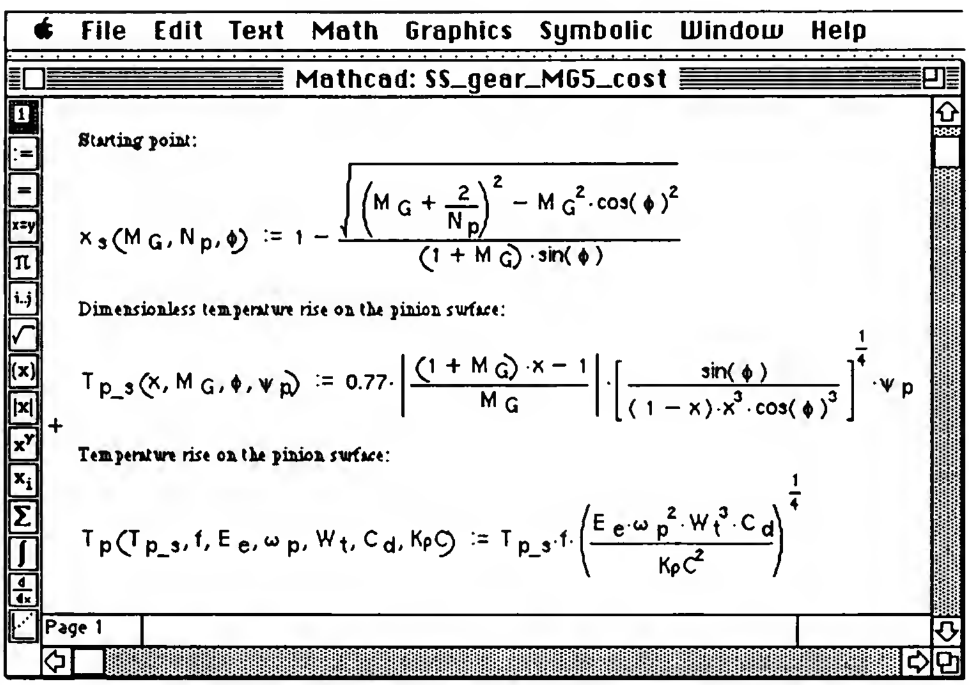


Figure 8-2. Mathcad document illustrating the use of standard math notation.

how the defined functions are displayed in standard math notation. Figure 8-3 demonstrates Mathcad's ability to deal with units. This feature is indispensable when working

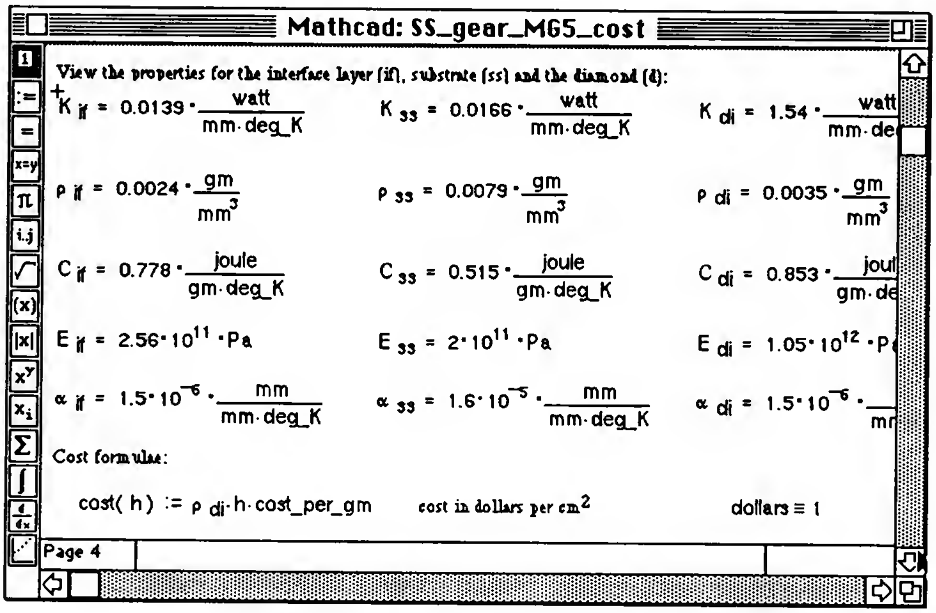


Figure 8-3. Mathcad document illustrating its capability for handling built-in units.

with a large number of parameters from a mix of English and metric units. Conversions are handled transparently by the system. Plotting capabilities are limited, but effective. Figure 8-4 shows a sample plot.

In the prototype system, the match is evaluated by building all the pertinent equations for both the discovery and the application directly in Mathcad. The design parameters are entered and simulation is run. Generated performance and cost data is exported from the Mathcad environment for use in the cost/benefit analysis module.

8.1.2 Cost/Benefit Analysis

Cost/benefit analysis is handled by post-processing cost and performance characteristics generated by the algorithmic module. The goal of this analysis is to find a functional

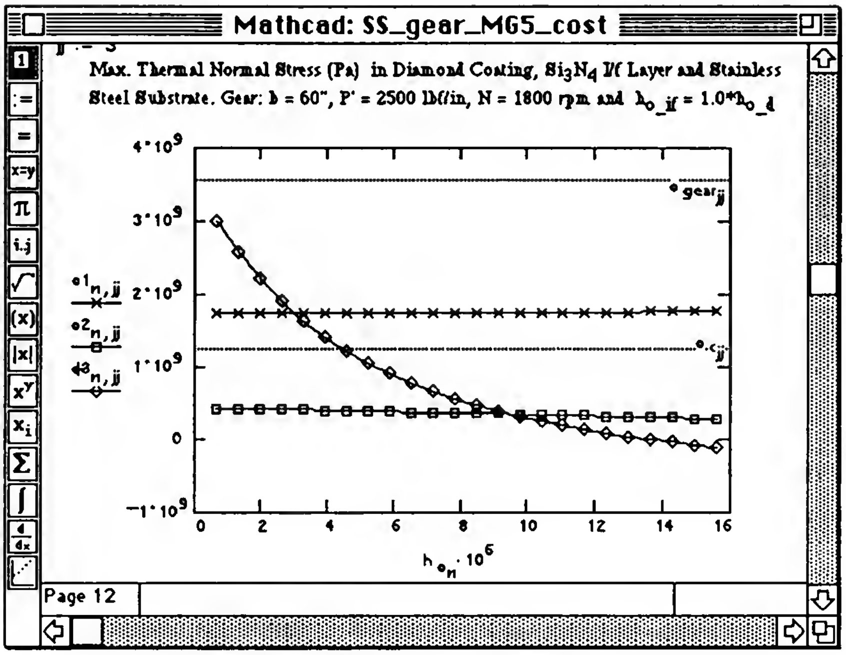


Figure 8-4. Mathcad document illustrating plotting capability.

relationship between cost and each performance characteristic. In this study, the performance characteristic most important is fatigue life extension. Cost and life figures from many simulations were exported to text files. These files were imported into a series of Excel (Microsoft, Inc.) spreadsheets for further data manipulation and plotting. Figure 8-5 illustrates the process of transferring data from one system to the next.

8.1.3 Ranking of Applications

Once the all discovery-application pairs have been evaluated for cost/benefit, it is a straightforward process to rank the applications from best to worst. If the applications share common objectives, such as life extension, then the ranking can be produced by inspection. If the performance criteria are dissimilar, then it is necessary to normalize the results to make a good judgement.

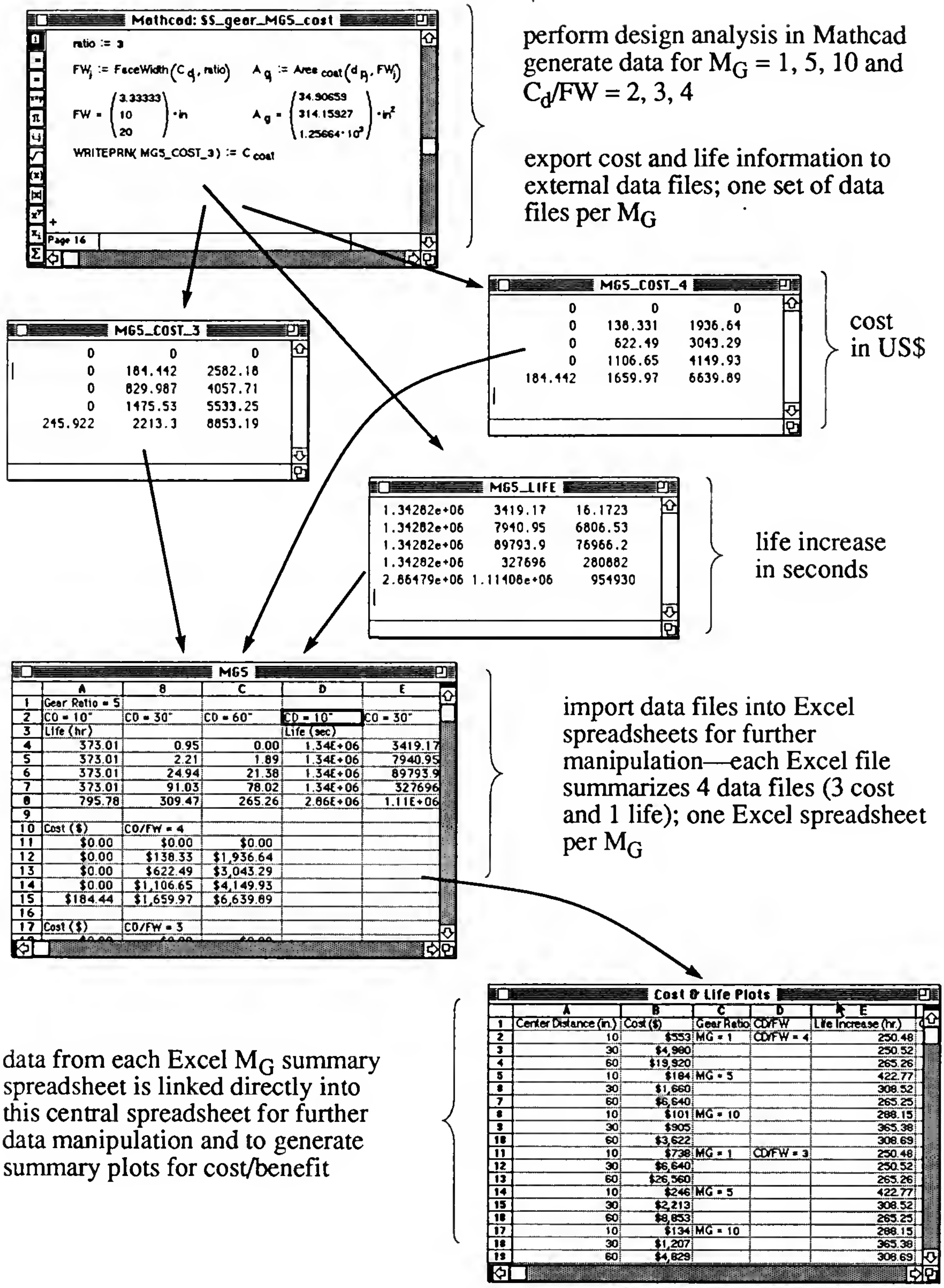


Figure 8-5. Data flow for the quantitative analysis procedure.

8.2 Application of the Quantitative Matching

The following example is based on the discovery–application match between the CVD diamond coating process and a standard dedendum gear pair as reported in the previous chapter. Results of the qualitative matching process indicated gears were a good technology match for the diamond CVD discovery. The gear pairs explored here are taken from relatively short life applications, such as rocket motors, where the gears must operate at extreme loading conditions. Uncoated gears under the same operating conditions would fail within very few cycles.

8.2.1 Qualitative Analysis Module Inputs

Relationships developed earlier for gear pairs were entered into a Mathcad document. Figure 8-2 shows a few of the equations entered for gear pair design. Next, empirical relationships developed for multilayer coating were entered as shown in Figure 8-6.

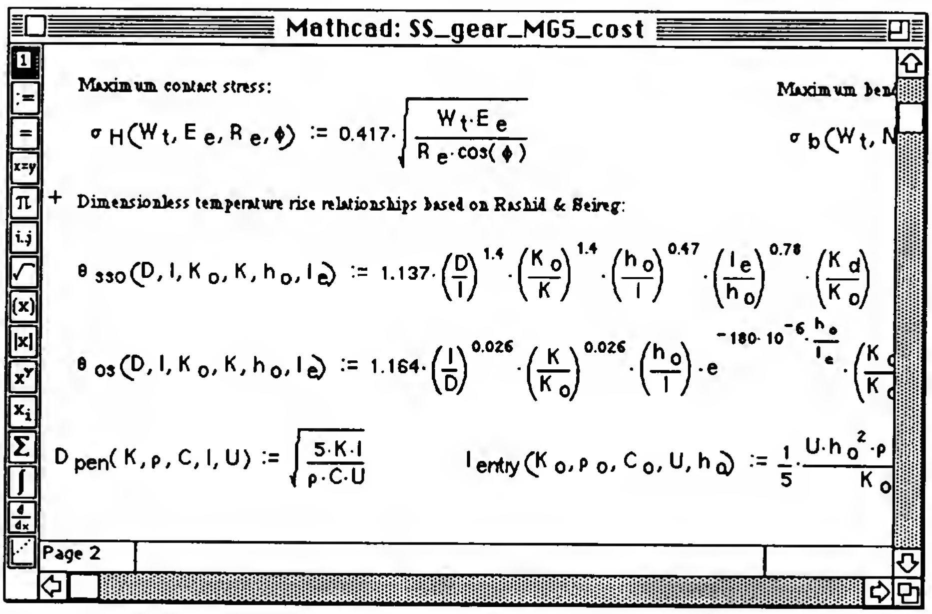


Figure 8-6. Empirical multilayer dimensionless relationships within Mathcad document

Equations referenced in the knowledge base were first arranged in a format suitable for designing gears. The user then supplied information necessary to bound the problem. It was determined that stainless steel gears for short duration applications—such as rocket motors—would be examined in this example. The design space inputs are given in Table 8-1. Note that the units are mixed in this table. One feature of the Mathcad software is the

Table 8-1. Design parameters for technology match

Match: Diamond-CVD + Gears	
Standard Addendum	
Parameter	**Value**
gear material	AISI 304 stainless steel
W_t	= 2500 lbf/in
ω_p	= 1800 rpm
f	= 0.045
ψ_p	= 0.5
ϕ	= 20°
σ_{bmax}	= 200,000 psi
K	= 0.1465 W/mm-K
ρ	= 0.0102 gm/mm^3
c	= 0.2554 J/gm-K
α	= 4.9 x 10^{-6} mm/mm-K
E_e	= 15,000,000 psi

ability to automatically convert units for consistent results. Bounding parameters for the design space are given in Table 8-1. A value of $200/gram was used for estimating cost for applying the diamond.

The range of values for the bound parameters will allow the designer to judge in which regimes that the new technology has the most merit and where it is wise to stay with existing technology.

Table 8-2. Design bounding parameters for technology match

Match: Diamond-CVD + Gears	
Standard Addendum	
Parameter	**Values**
C_d	= 10", 30", 60"
M_G	= 1, 5, 10
C_d/FW	= 2, 3, 4

For each value of the gear ratio, M_G, a new Mathcad document was created. The document "SS_gear_MG5_cost" contains the design simulation for stainless steel gear pairs of gear ratio 5. Likewise "SS_gear_MG1_cost" and "SS_gear_MG10_cost" contain simulation information for stainless steel gear pairs of gear ratio 1 and 10, respectively. Each document calculates life increase and cost information for using the diamond technology on gear pair design. The life and cost information is written to external data files for use by Excel spreadsheets that perform the final cost/benefit analysis. The procedure is illustrated in Figure 8-5 for the file "SS_gear_MG5_cost."

8.2.2 Results

The normalized coating cost per hour of life increase is shown in Figures 8-7, 8-8 and 8-9. The results indicate that for low gear ratios and large center distance that diamond coating is not cost effective.

Increase in pinion life is summarized in Figures 8-10, 8-11 and 8-12. All pinion combinations appear to gain significant life increase due to the diamond coating.

The cost of coating the pinion gears using cost parameters available today is summarized in Figures 8-13, 8-14 and 8-15. The coating cost increases as the center distance increases and the facewidth increases. This simply means that as the size of the pinion increases, it takes more material to coat it and thus more cost is involved.

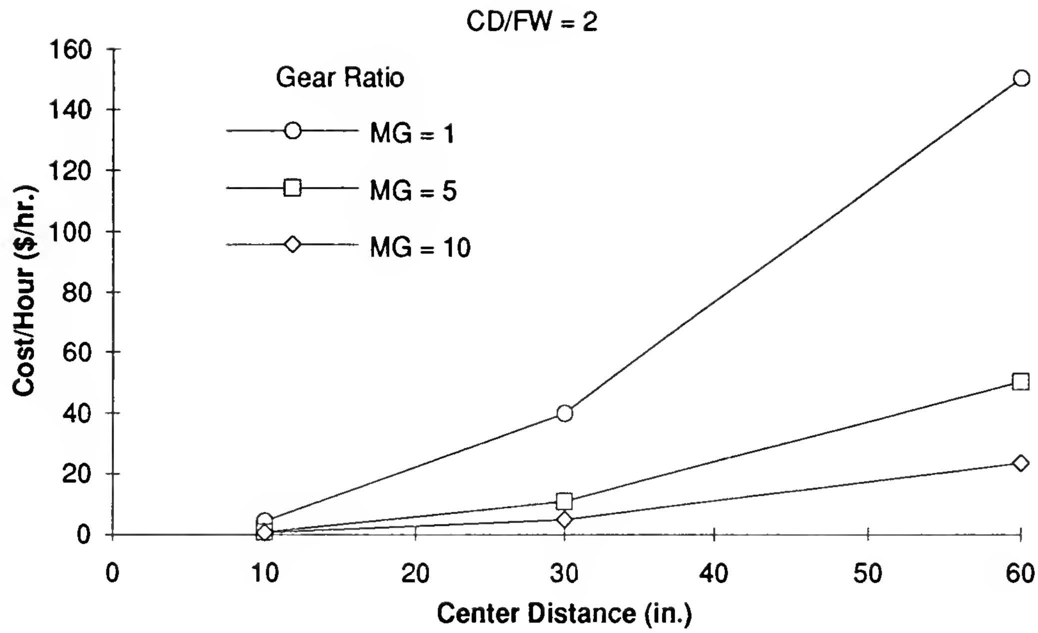


Figure 8-7. Coating cost normalized to increase in life in the pinion for three gear ratios holding the ratio of center distance to facewidth at 2:1.

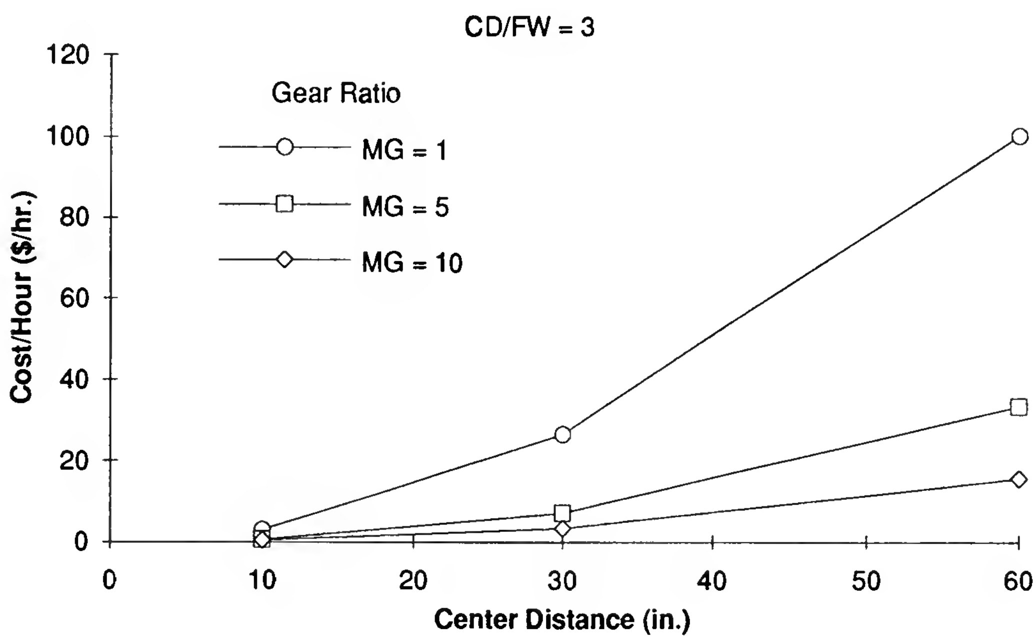


Figure 8-8. Coating cost normalized to increase in life in the pinion for three gear ratios holding the ratio of center distance to facewidth at 3:1.

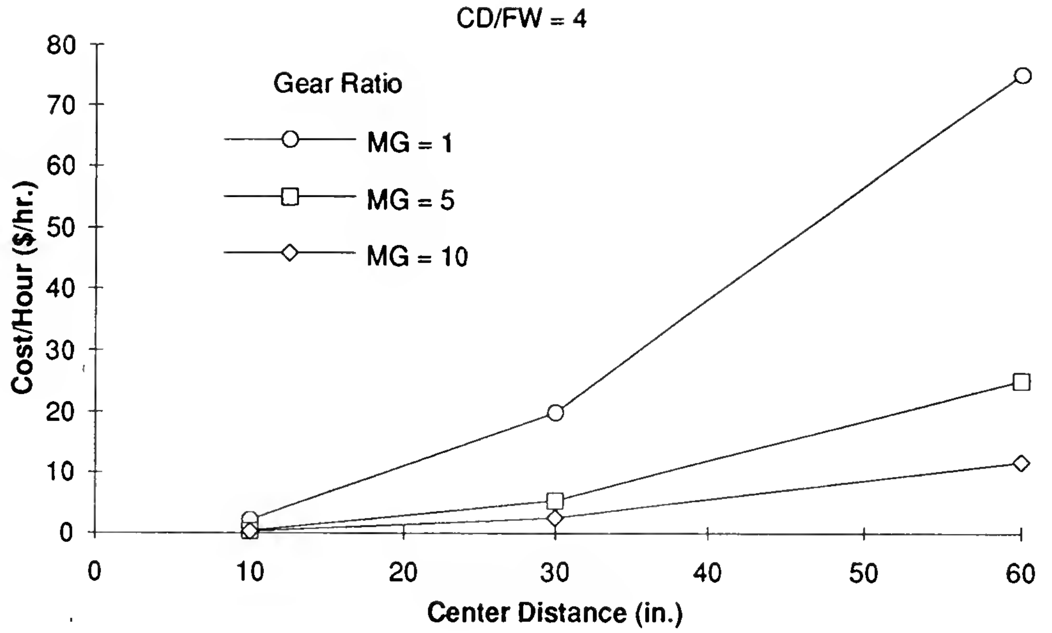


Figure 8-9. Coating cost normalized to increase in life in the pinion for three gear ratios holding the ratio of center distance to facewidth at 4:1.

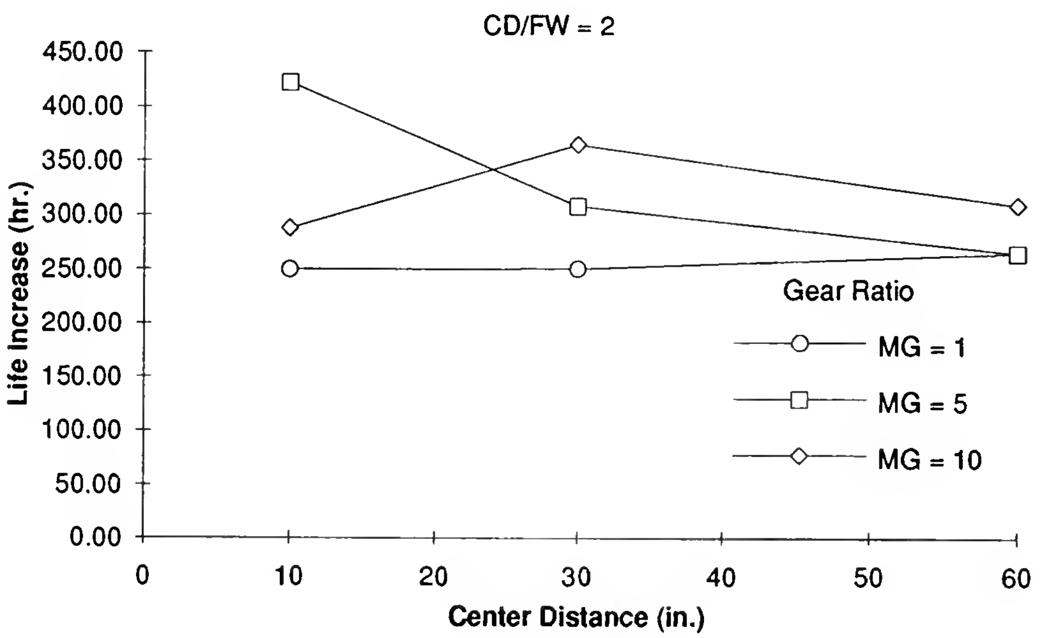


Figure 8-10. Increase in life in the pinion for three gear ratios holding the ratio of center distance to facewidth at 2:1.

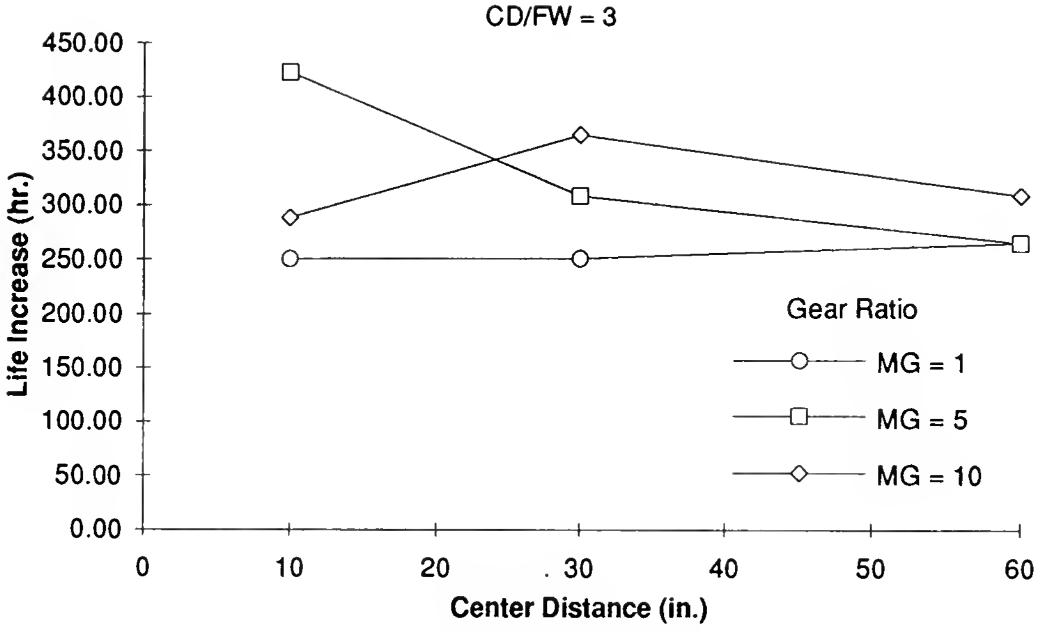


Figure 8-11. Increase in life in the pinion for three gear ratios holding the ratio of center distance to facewidth at 3:1.

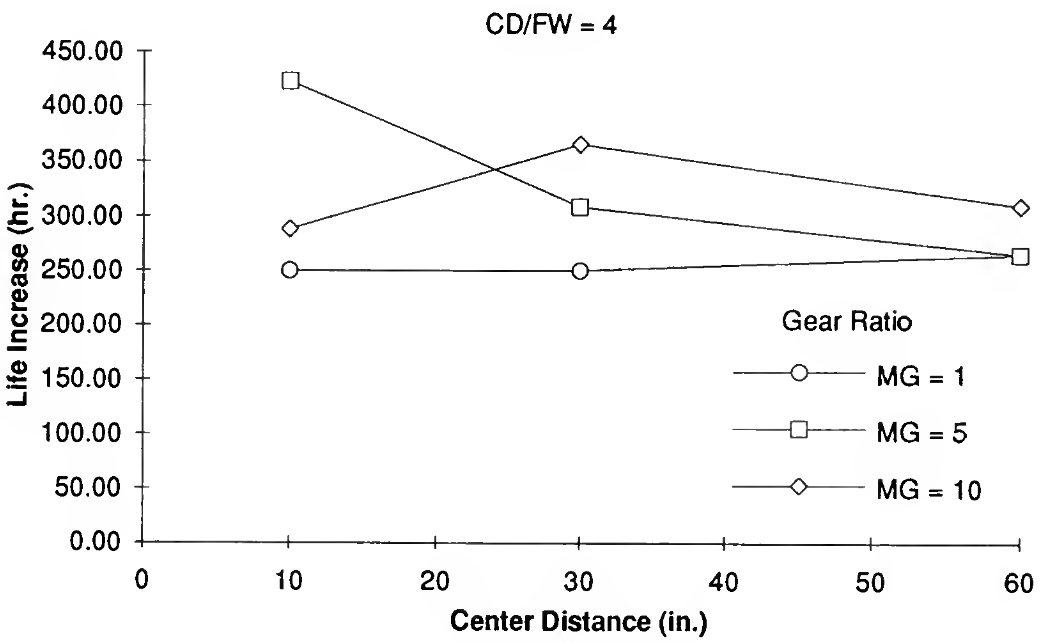


Figure 8-12. Increase in life in the pinion for three gear ratios holding the ratio of center distance to facewidth at 4:1.

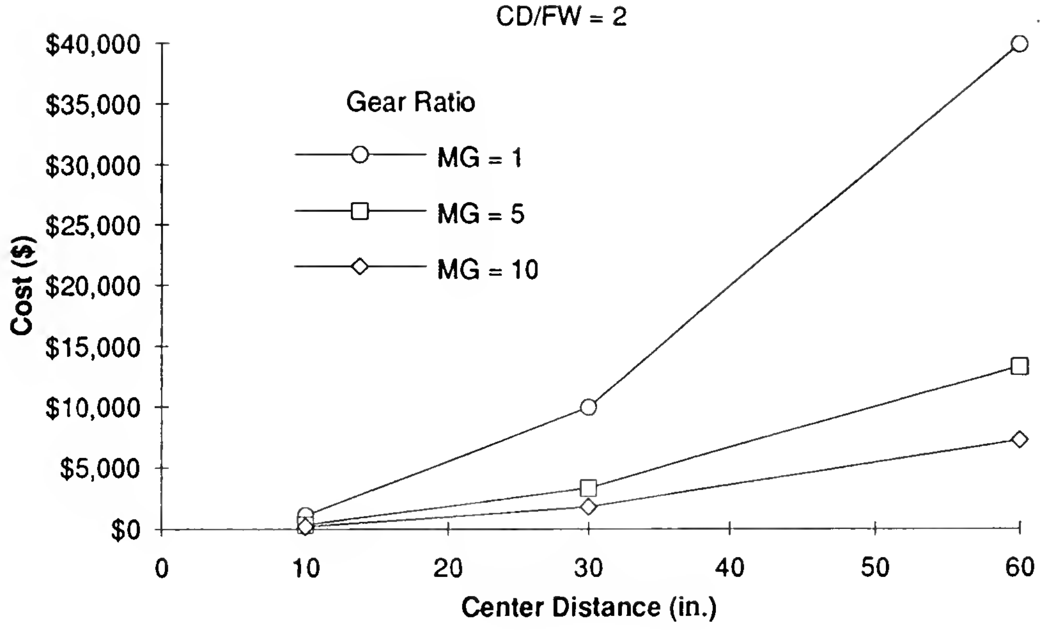


Figure 8-13. Coating cost for the pinion to meet full life for three gear ratios holding the ratio of center distance to facewidth at 2:1.

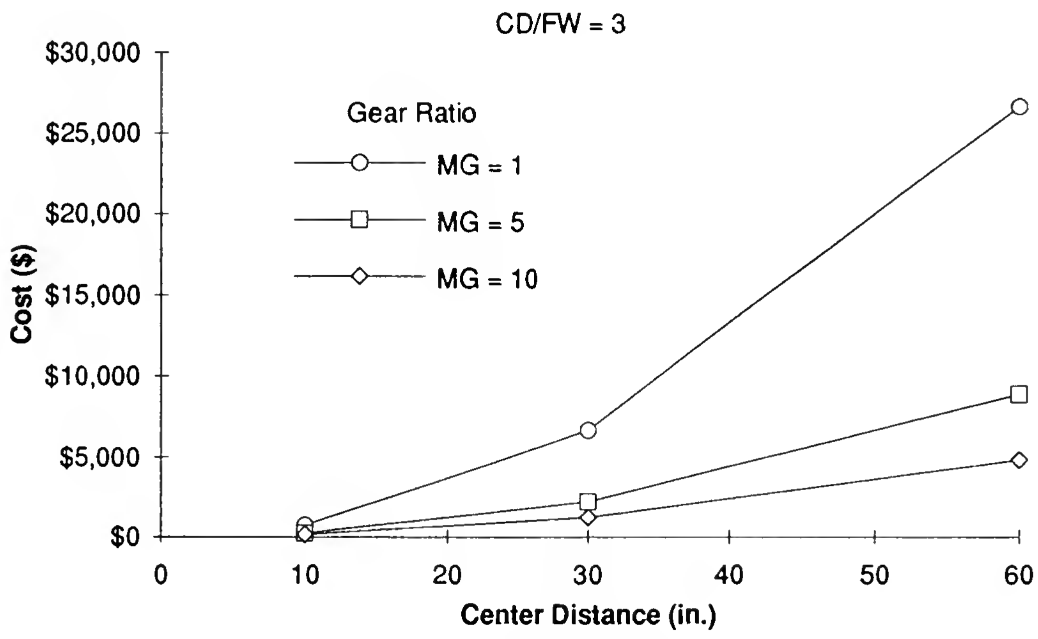


Figure 8-14. Coating cost for the pinion to meet full life for three gear ratios holding the ratio of center distance to facewidth at 3:1.

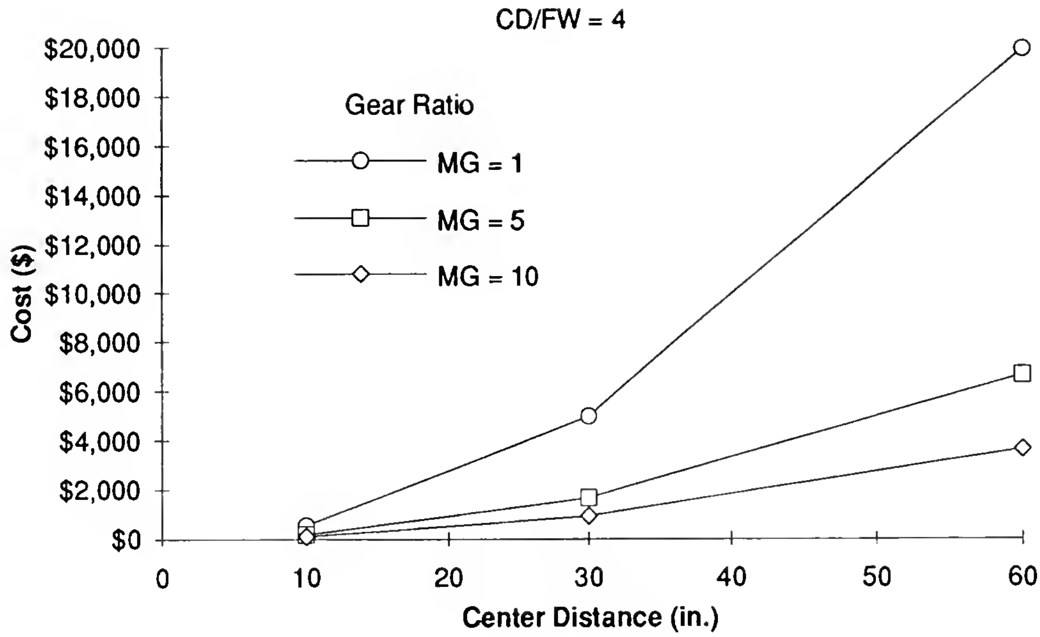


Figure 8-15. Coating cost for the pinion to meet full life for three gear ratios holding the ratio of center distance to facewidth at 4:1.

CHAPTER 9
SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

9.1 Summary

The study reported in this dissertation can be summarized as follows:

1. A framework for capturing the attributes for a new technological innovation and its potential applications is developed. The framework is built around an object-oriented expert system knowledge base. Knowledge about the technology is incorporated via objects, facts, rules and procedural functions. The framework is represented in a hierarchical object inheritance tree. The attributes for potential applications are encoded in the same knowledge base and are organized to facilitate matching with technological discoveries.

2. A two-level matching process is proposed. The first level is qualitative matching for any potential applications that might fit a new technological discovery. This matching process occurs within the context of a knowledge-based system. Qualitative needs for a particular application are decomposed into individual property attributes. These attributes are then compared to attributes within the technological discovery's capabilities. Application–discovery matches are then passed on to the next level for quantitative analysis. The second level is a quantitative procedure for evaluating the cost-benefit of a specific application of the new technology. The procedure performs a design optimization process to minimize the manufacturing cost of the application.

1. A proposed two-level matching algorithm is evaluated for the ability to rank potential applications in order of highest promise for further technology development. The algorithm is demonstrated by matching applications to a diamond coating discovery. A methodology is developed for incorporating diamond coating into the design process for sliding, concentrated contact applications. These applications can include, among others, gears, bearings, traction drives, brakes and clutches.

9.2 Conclusions

The following conclusions can be made based on the work reported in this dissertation:

1. It is feasible to develop a computer-based methodology for evaluation of the potential applications of new technology. The degree of qualitative matching and quantitative

cost-benefit results allow potential applications to be ranked on the basis of highest promise for further technology development.

2. The heuristic matching process developed is extremely flexible due to the objected-oriented design principles incorporated in its structure. The implementation of the heuristic matching process using an object-oriented knowledge base management system is further enhanced by the use of the Rete multi-attribute pattern matching algorithm. The process, due to its extensible architecture, is capable of handling larger problem domains.

3. The quantitative analysis procedure provides a flexible method of evaluating the cost/benefit for a particular application of new technology. The current process is very time intensive on the user. This process can be streamlined through the use of a more robust evaluation environment. This environment should include optimization modules to further automate the methodology.

4. It is shown in the example of surface coating of gears for short duration applications, that significant improvements can be made in the durability of stainless steel components.

5. It is also shown in cases where sliding is not too high, such as highly loaded, well-lubricated roller bearings, that the coating is not economically viable. In these cases thermal effects are minimal and current technology is adequate.

9.3 Recommendations

The following is recommended for future studies:

1. Before the algorithm is generalized, it should be tested on other laboratory developments to modify it if necessary to insure its generality.

2. An attribute weighting scheme should be developed to emphasize critical attributes in the qualitative process. For instance, in the example gear application, an attribute such as thermal conductivity should be weighted heavier than Poisson's ratio. In this case, a strong Poisson's ratio match would not cancel out a weak thermal conductivity match. Likewise, a strong thermal conductivity match will dominate one or more weak matches in less heavily weighted attributes.

3. Design optimization capability should be added to the quantitative portion of the developed methodology.

4. An objective function class should be added to the governing equations class within the attribute hierarchy tree.

5. The qualitative and quantitative modules should be tightly integrated under one application framework.

6. To improve efficiency, an object-feature, numerical coding scheme should be considered for representing discovery and potential application characteristics.

7. A more sophisticated object-oriented knowledge base management system should be considered for the qualitative attribute matching procedure. The system should be capable of handling complex attribute interactions.

8. The cost/benefit analysis system should be improved to handle a variety of performance characteristics. For example, two applications for a discovery may have different performance characteristics—such as life improvement and cycle time—compared to cost. The benefits for these characteristics should be pushed up to a higher level where they can be compared on the same basis. A ranking judgement based on return on research investment could provide this capability.

9. When the technology knowledge bases contain few applications and discoveries, paper studies are more time and cost effective approaches for the matching process. At some technology knowledge base population level a "critical mass" is reached. Once the knowledge bases exceed this saturation level, it is more cost effective to use the developed process. An estimate of the "critical mass" knowledge base population level should be determined. This estimate will provide a basis for determining a payoff for the developed matching process.

10. Once the "critical mass" technology knowledge base saturation level (as mentioned previously) is attained, the issue of spontaneous insight[1] can be approached. It is possible for the proposed methodology to make a divergent leap into an unanticipated application domain if a wide variety of application domains are encoded. This capability should be evaluated once the system reaches "critical mass." The burden is on the knowledge base developer to encode a diverse sampling of potential applications.

11. A large scale distribution plan should be developed to disseminate the results. The Internet computer network might provide the basis for a public-access technology evaluation system.

12. The two-level matching process—using diamond coating as the discovery—should be checked for other applications such as bearings, overrunning clutches and brakes.

13. Based on the different case studies, a general, fully automated system should be developed for this particular process.

14. The surface treatment and diamond film coating on selected metals should be characterized for fatigue impact.

---

1. An example of spontaneous insight would be the unexpected use of lasers as pointing devices for presenters.

15. The transient multilayer coating thermal model developed should be verified via a finite difference or finite element analysis. The analysis should be verified experimentally if practical.

APPENDIX A
RELEVANT TRIBOLOGICAL EQUATIONS

The equations in this appendix summarize relevant tribological equations for the prediction of temperature rise and the prediction of coefficient of friction. References that are not listed in the References section were provided in course notes from EML 5505.

A.1 Thermal Considerations

A.1.1 Heat partition and transient temperature effects

Relationships for layered tribological surfaces as developed in "Heat Partition and Transient Temperature Distribution in Layered Concentrated Contacts [Rash86]."

1. Heat Source Moving Over a Semi-Infinite Solid

$$T_S - T_B = 1.123\frac{q_t}{l}\sqrt{\frac{l}{K\rho CU}} \tag{A.1}$$

$$\frac{(T_S - T_B)K}{q_t} = 1.03\left(\frac{\rho CUl}{K}\right)^{-0.5} \tag{A.2}$$

2. Sliding/Rolling Dry Contacts

$$\alpha = \frac{1}{1 + \sqrt{\dfrac{\rho_2 C_2 U_2 K_2}{\rho_1 C_1 U_1 K_1}}} \tag{A.3}$$

$$T_{S1} - T_{B1} = \frac{q_t}{\sqrt{l}}\frac{1.03}{\sqrt{K_1\rho_1 C_1 U_1} + \sqrt{K_2\rho_2 C_2 U_2}} \tag{A.4}$$

3. Heat Source With a Hertzian Distribution Moving Over Layered Semi-Infinite Solid

$$\frac{(T_S - T_{S0})K_0}{q_t} = 1.137\left(\frac{D}{l}\right)^{1.4}\left(\frac{h_0}{l}\right)^{0.47}\left(\frac{l_e}{h_0}\right)^{0.78} \tag{A.5}$$

$$\frac{(T_0 - T_S)K_0}{q_t} = 1.164\left(\frac{l}{D}\right)^{0.026}\left(\frac{K}{K_0}\right)^{0.026}\left(\frac{h_o}{l}\right)e^{-180\times10^{-6}\frac{l_e}{h_0}}$$

$$D = \sqrt{\frac{5Kl}{\rho CU}} \tag{A.6}$$

$$l_e = \frac{1}{5}\frac{Uh_0^2\rho_0C_0}{K_0}$$

4. Lubricated Rolling/Sliding Contacts

$$\frac{(T_{01} - T_{S1})}{\alpha q_t} = B_1$$

$$\frac{(T_{02} - T_{S2})}{(1-\alpha)q_t} = B_2$$

$$B_1 = \frac{1.14}{K_0}\left[\frac{U_1\rho_1C_1h}{2K_0}\right]^{0.013}\left(\frac{2l}{h}\right)^{-1.003}\left(\frac{K_1}{K_0}\right)^{0.013}e^{-900\times10^{-6}\left(\frac{U_1\rho_0C_0h}{2K_0}\right)}$$

$$B_2 = \frac{1.14}{K_0}\left[\frac{U_2\rho_2C_2h}{2K_0}\right]^{0.013}\left(\frac{2l}{h}\right)^{-1.003}\left(\frac{K_2}{K_0}\right)^{0.013}e^{-900\times10^{-6}\left(\frac{U_2\rho_0C_0h}{2K_0}\right)} \tag{A.7}$$

$$\frac{(T_{S1} - T_{B1})}{\alpha q_t} = \frac{1.03}{K_1}\left(\frac{\rho_1C_1U_1l}{K_1}\right)^{-05} = A_1$$

$$\frac{(T_{S2} - T_{B2})}{(1-\alpha)q_t} = \frac{1.03}{K_2}\left(\frac{\rho_2C_2U_2l}{K_2}\right)^{-05} = A_2$$

$$T_{01} = T_{02}$$

$$\alpha = \frac{1}{A_1 + B_1 + A_2 + B_2}\left(\frac{(T_{B2} - T_{B1})}{q_t} + A_2 + B_2\right)$$

$$T_{S01} - T_{B1} = \gamma_1 q_t A_1$$
$$T_{S02} - T_{B2} = \gamma_2 q_t A_2 \tag{A.8}$$

$$T_{S01} - T_{B1} = \alpha q_t [A_1 - c_1]$$

$$T_{S02} - T_{B2} = (1-\alpha) q_t [A_2 - c_2]$$

$$c_1 = \frac{1}{K_0}\left(\frac{U_1\rho_1 C_1 h}{2K_0}\right)^{-0.7}\left(\frac{2l}{h}\right)^{-1.17}\left(\frac{K_1}{K_0}\right)^{-0.7}\left(\frac{U_1\rho_0 C_0 h}{2K_0}\right)^{0.78}$$

$$c_2 = \frac{1}{K_0}\left(\frac{U_2\rho_2 C_2 h}{2K_0}\right)^{-0.7}\left(\frac{2l}{h}\right)^{-1.17}\left(\frac{K_2}{K_0}\right)^{-0.7}\left(\frac{U_2\rho_0 C_0 h}{2K_0}\right)^{0.78} \tag{A.9}$$

$$\gamma_1 = \alpha\frac{(A_1 - c_1)}{A_1}$$

$$\gamma_2 = (1-\alpha)\frac{(A_2 - c_2)}{A_2}$$

$$T_{01} - T_B = \alpha q_t A_1 + \alpha q_t B_1 e^{-0.5\left[\frac{w}{h}\right]} \tag{A.10}$$

$$T_{01} = T_3$$

$$H_{min} = 3.07 U_0^{0.71} G_0^{0.57} P_0^{-0.11}$$

$$dH_{min} = \frac{h}{R_e}, P_0 = \frac{W_o}{E_e R_e}, U_0 = \frac{\eta_0 UR}{E_e R_e}, G_0 = \alpha_v E_e \tag{A.11}$$

$$H_e = CC_0 H_{min}$$

$$CC_0 = [1 + 0.241((1 + 14.8Z^{0.83})L^{0.64})]^{-1}$$

$$Z = (U_1 - U_2)/(U_1 + U_2) \tag{A.12}$$

$$L = \text{Thermal loading parameter} = (\eta_0\beta(UR)^2)/K_3$$

$$h = H_c R_e \tag{A.13}$$

5. A Parabolic Heat Source Moving on a Metallic Semi-Infinite Solid With a Surface Layer of Low Conductivity

$$IC = \frac{1.085}{K_0}\left[\frac{U\rho C h_0}{K_0}\right]^{-0.795}\left[\frac{l}{h_0}\right]^{-1.07}\left[\frac{K}{K_0}\right]^{-0.77}\left[\frac{U\rho_0 C_0 h_0}{K_0}\right]^{0.763}$$

$$\frac{(T_S - T_{SO})}{q_t} = IC \tag{A.14}$$

$$IB = \frac{0.135}{K_0}\left[\frac{U\rho C h_0}{K_0}\right]^{-0.21}\left[\frac{l}{h_0}\right]^{-0.85}\left[\frac{K}{K_0}\right]^{-0.25} e^{\left(-5750\times10^{-6}\left[\frac{U\rho_0 C_0 h_0}{K_0}\right]\right)} \tag{A.15}$$

$$\frac{(T_0 - T_S)}{q_t} = IB$$

6. A Parabolic Heat Source Moving on a Low Conductivity Semi-Infinite Solid With a Metallic Surface Layer

$$CC = \frac{0.363}{K_0}\left[\frac{U\rho C h_0}{K_0}\right]^{-0.831}\left[\frac{l}{h_0}\right]^{-0.857}\left[\frac{K}{K_0}\right]^{-0.836}\left[\frac{U\rho_0 C_0 h_0}{K_0}\right]^{0.627} \tag{A.16}$$

$$\frac{(T_S - T_{S0})}{q_t} = CC$$

$$CB = \frac{0.363}{K_0}\left[\frac{U\rho C h_0}{K_0}\right]^{-0.98}\left[\frac{l}{h_0}\right]^{-0.84}\left[\frac{K}{K_0}\right]^{-1}\left[\frac{U\rho_0 C_0 h_0}{K_0}\right]^{0.8} \tag{A.17}$$

$$\frac{(T_0 - T_S)}{q_t} = CB$$

7. Dry Layered Contacts

$$\alpha = \frac{1}{A_1 + Z_1 + A_2 + Z_2}\left[\frac{(T_{B2} - T_{B1})}{q_t} + A_2 + Z_2\right] \tag{A.18}$$

$$T_{S01} - T_{B1} = \alpha q_t [A_1 - Z_3]$$
$$T_{S02} - T_{B2} = (1-\alpha) q_t [A_2 - Z_4] \tag{A.19}$$

$$\gamma_1 = \alpha([A_1 - Z_3]/A_1)$$
$$\gamma_2 = (1-\alpha)([A_2 - Z_4]/A_2) \tag{A.20}$$

$$T_{01} - T_{B1} = \alpha q_t [A_1 + Z_1]$$
$$T_{01} = T_{02} \tag{A.21}$$

A.2 Frictional Considerations

A.2.1 Friction factors for high slide to roll ratios

Misharin:

$$f_t = \frac{0.325}{(\nu_0 u v)^{1/4}} \tag{A.22}$$

$$0.02 < f_t < 0.08$$

Benedict and Kelly:

$$f = 0.0127\log\left|\frac{3.17\times10^8 W}{\mu_0 u^2 v}\right| \tag{A.23}$$

Cameron:

$$f = \frac{S+22}{35}\frac{0.6}{\mu^{1/8}v^{1/3}u^{1/6}R^{1/2}} \tag{A.24}$$

Kelly and Lemanski:

$$f = 0.0099\log\left[\frac{3.89\times10^7 W}{\mu_0 v u^2\left(\frac{R_1+R_2}{3}\right)^2}\right] \tag{A.25}$$

Drozdov and Gavrikov:

$$f = \frac{1}{0.8\nu_0^{1/2}v + u\phi(P_{max}, \nu_0) + 13.4} \tag{A.26}$$

$$\phi(P_{max}, \nu_0) = 0.47 - 0.13\times10^4 P_{max} - 0.4\times10^{-3}\nu_0$$

$$f = C\left[\frac{SW}{\eta_0 R v_T}\right]^{1/4} \tag{A.27}$$

A.2.2 Effect of lubricant properties on temperature and wear in sliding concentrated contacts

Refer to [Seir81].

A.2.3 Friction and temperature in rolling sliding contacts

Refer to [O'Do66].

$$f = \left(\frac{S+22}{35}\right)\frac{0.6}{\Im}$$

$$\Im = \mu^{1/8}(V_1 - V_2)^{1/3}(V_1 + V_2)^{1/6}R^{1/2} \tag{A.28}$$

$$f = \left(\frac{S+22}{35}\right)\frac{1.5}{(\Im)_{\mathrm{cm}}}$$

A.2.4 Wear Resistance in Brake Materials

Wear in brakes:

$$\text{wear resistence} \propto \frac{\sigma_{\mathrm{ult}}(1-\mu)\,(C\rho K^3)^{1/4}}{E\alpha} \tag{A.29}$$

Thin surfaces:

$$\text{wear resistence} \propto \frac{\sigma(1-\mu)\sqrt{K\rho C}}{E\alpha} \tag{A.30}$$

Resistance to thermal shock:

$$\text{thermal shock resistence} \propto \frac{K\sigma_0}{E\alpha} \tag{A.31}$$

A.2.5 Elastohydrodynamics

Grubin (1949), elastic and variable viscosity:

$$\frac{h_0}{R} = \frac{1.95\,(\eta_0\mu\alpha)^{8/11}E^{1/11}}{R^{7/11}P_y^{1/11}}$$

$$\frac{1}{E'} = \frac{1}{2}\left[\frac{1-\sigma_1^2}{E_1} + \frac{1-\sigma_2^2}{E_2}\right] \tag{A.32}$$

$$\frac{h_0\,(\text{elastic and variable viscosity})}{h_0\,(\text{rigid and constant viscosity})} = 0.398\frac{\alpha^{8/11}E'^{1/11}P_y^{10/11}}{R^{7/11}(\eta_0\mu)^{3/11}}$$

A.2.6 Thermohydrodynamic Shear Zone Thickness

Seireg (1994):

$$\frac{W}{h} = \frac{1}{5\pi}\left[1 - \mathrm{e}^{-\left(\frac{Y - 1/(5\pi)}{2/\pi}\right)}\right] \tag{A.33}$$

$$Y = (SP, 10^{6})$$

APPENDIX B
PROGRAM LISTINGS

B.1 BATCH FILE

technology KBS.bat

```
(clear)
; define templates for match table tracking
(load "Match Templates.CLP")
; define class hierarchy for the new technology knowledge base
(load "new technology tree.CLP")
(load "Governing Equations.CLP")
(load "qualitative needs.CLP")
(load "Match Table.CLP")
(load "Match Tally Class.CLP")
; define instances of the technology knowledge base attributes
(load "CVD-Diamond instances.CLP") ; DISCOVERY
(load "CVD-Diamond Equations.CLP")
(load "Gear Instances.CLP") ; APPLICATION
(load "Gear Equations.CLP")
(load "Hi-Load-Brg Instances.CLP")
; define generic methods
(load "Generic Methods.CLP")
; define functions
(load "Matching Functions.CLP")
(load "Menu System.CLP")
(load "File Functions.CLP")
; define rules
(load "Matching Rules.CLP")
(load "Qualitative Rules.CLP")
(load "Menu Rules.CLP")
; make sure we can always find the help file
(help-path "Keith's PPC:Development:CLIPS:CLIPS 6.0:Executables:
clips.hlp")
(reset)
;(clear-window)
;
;=====================================================================
;
; Welcome to the Qualitative Technology Discovery--
; Application Matcher v.1.0 (4/20/95)
```

```
;
; To begin the search process, enter (run) and
; press <return>
;
;==================================================================
```

B.2 KNOWLEDGE BASE STRUCTURE

B.2.1 Class Definitions

new technology tree.CLP

```
(defclass TECHNOLOGY
   "TECHNOLOGY is the top level object of the new technology knowledge
    base"
   (is-a USER)
   (role concrete)
   (pattern-match reactive)
   (slot technology-name
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type LEXEME)
      (default ?NONE))
   (slot technology-type
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type LEXEME)
      (default DISCOVERY)
      (allowed-values DISCOVERY APPLICATION))
   (slot phylum
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type LEXEME)
      (default ?NONE))  ;technology  code
)   ;end defclass TECHNOLOGY

(defclass LEVEL-1-ATTRIBUTE
   "LEVEL-1-ATTRIBUTE is an abstract class that describes the
    first-level attributes of a technology's property"
   (is-a USER)
   (role abstract)
   (pattern-match non-reactive)
   (slot classification
      (create-accessor read-write)
      (access read-write)
```

```
      (type LEXEME)
      (default ?DERIVE))
)   ; end defclass LEVEL-1-ATTRIBUTE

(defclass LEVEL-2-ATTRIBUTE
   "LEVEL-2-ATTRIBUTE is an abstract class that describes the
    second-level attributes of a technology's property"
   (is-a USER)
   (role abstract)
   (pattern-match non-reactive)
   (slot order
      (create-accessor read-write)
      (access read-write)
      (type LEXEME)
      (default ?DERIVE))
)   ; end defclass LEVEL-2-ATTRIBUTE

(defclass LEVEL-3-ATTRIBUTE
   "LEVEL-3-ATTRIBUTE is an abstract class that describes the
    third-level attributes of a technology's property"
   (is-a USER)
   (role abstract)
   (pattern-match non-reactive)
   (slot family
      (create-accessor read-write)
      (access read-write)
      (type LEXEME)
      (default ?DERIVE))
)   ; end defclass LEVEL-3-ATTRIBUTE

(defclass LEVEL-4-ATTRIBUTE
   "LEVEL-4-ATTRIBUTE is an abstract class that describes the
    fourth-level attributes of a technology's property"
   (is-a USER)
   (role abstract)
   (pattern-match non-reactive)
   (slot species
      (create-accessor read-write)
      (access read-write)
      (type LEXEME)
      (default ?DERIVE))
   (slot assessment-criteria
      (create-accessor read-write)
      (access read-write)
      (type LEXEME)
      (allowed-values
                    NEED
                    DESIRABLE-FEATURE
                    OPTIONAL-FEATURE
                    CAPABILITY)
      (default CAPABILITY))
)   ; end defclass LEVEL-4-ATTRIBUTE

(defclass TECHNOLOGY-PROPERTY
```

```
   "TECHNOLOGY-PROPERTY is an abstract class that describes the
    qualitative and quantitative characteristics of a technology's
    property"
   (is-a USER)
   (role abstract)
   (pattern-match non-reactive)
   (slot qualitative-value
      (create-accessor read-write)
      (access read-write)
      (type LEXEME)
      (allowed-values
                     VERY-LOW
                     LOW
                     MEDIUM-LOW
                     MEDIUM
                     MEDIUM-HIGH
                     HIGH
                     VERY-HIGH))
   (slot qualitative-value-range
      (create-accessor read-write)
      (access read-write)
      (type LEXEME)
      (range ?VARIABLE ?VARIABLE)
      (default ?DERIVE))
   (slot quantitative-value
      (create-accessor read-write)
      (access read-write)
      (type NUMBER)
      (default ?DERIVE))
   (slot quantitative-value-range
      (create-accessor read-write)
      (access read-write)
      (type NUMBER)
      (range ?VARIABLE ?VARIABLE)
      (default ?DERIVE))
   (slot units
      (create-accessor read-write)
      (access read-write)
      (type LEXEME)
      (default ?DERIVE))
   (slot symbol
      (create-accessor read-write)
      (access read-write)
      (type LEXEME)
      (default ?DERIVE))
)   ;end defclass TECHNOLOGY-PROPERTY

(defclass MFG-PROCESS-METHOD-LIST
   "MFG-PROCESS-METHOD-LIST is an abstract class that lists the steps involved
    in a particular manufacturing process"
   (is-a USER)
   (role abstract)
   (pattern-match non-reactive)
   (slot species
```

```
      (create-accessor read-write)
      (access read-write)
      (type LEXEME))
   (multislot method-list
      (create-accessor read-write)
      (access read-write)
      (type LEXEME))
)   ;end defclass MFG-PROCESS-METHOD-LIST

(defclass MATERIAL-ATTRIBUTE
   "MATERIAL-ATTRIBUTE is a first-level attribute specialization of the
    TECHNOLOGY attribute hierarchy"
   (is-a TECHNOLOGY LEVEL-1-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot classification
      (default MATERIAL))
)   ;end defclass MATERIAL

(defclass MANUFACTURING-ATTRIBUTE
   "MANUFACTURING-ATTRIBUTE is a first-level attribute specialization of
    the TECHNOLOGY attribute hierarchy"
   (is-a TECHNOLOGY LEVEL-1-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot classification
      (default MANUFACTURING))
)   ;end defclass MANUFACTURING-ATTRIBUTE

(defclass OTHER-ATTRIBUTE
   "OTHER-ATTRIBUTE is a first-level attribute specialization of
    the TECHNOLOGY attribute hierarchy"
   (is-a TECHNOLOGY LEVEL-1-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot classification
      (default OTHER))
)   ;end defclass OTHER-ATTRIBUTE

(defclass PHYSICAL-PROPERTY
   "PHYSICAL-PROPERTY is a SECOND-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MATERIAL superclass"
   (is-a MATERIAL-ATTRIBUTE LEVEL-2-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot order
      (default PHYSICAL))
)   ;end defclass PHYSICAL-PROPERTY

(defclass ECONOMIC-PROPERTY
   "ECONOMIC-PROPERTY is a SECOND-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MATERIAL superclass"
   (is-a MATERIAL-ATTRIBUTE LEVEL-2-ATTRIBUTE)
   (role concrete)
```

```
   (pattern-match reactive)
   (slot order
      (default ECONOMIC))
)   ;end defclass ECONOMIC-PROPERTY

(defclass PROCESS-PROPERTY
   "PROCESS-PROPERTY is a SECOND-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MANUFACTURING-ATTRIBUTE
superclass"
   (is-a MANUFACTURING-ATTRIBUTE LEVEL-2-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot order
      (default PROCESS))
)   ;end defclass PROCESS-PROPERTY

(defclass OTHER-PROPERTY
   "OTHER-PROPERTY is a SECOND-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MATERIAL superclass"
   (is-a MATERIAL-ATTRIBUTE LEVEL-2-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot order
      (default OTHER))
)   ;end defclass OTHER-PROPERTY

(defclass THERMAL-PROPERTY
   "THERMAL-PROPERTY is a THIRD-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PHYSICAL-PROPERTY superclass"
   (is-a PHYSICAL-PROPERTY LEVEL-3-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot family
      (default THERMAL))
)   ;end defclass THERMAL-PROPERTY

(defclass MECHANICAL-PROPERTY
   "MECHANICAL-PROPERTY is a THIRD-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PHYSICAL-PROPERTY superclass"
   (is-a PHYSICAL-PROPERTY LEVEL-3-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot family
      (default MECHANICAL))
)   ;end defclass MECHANICAL-PROPERTY

(defclass ENVIRONMENTAL-PROPERTY
   "ENVIRONMENTAL-PROPERTY is a THIRD-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PHYSICAL-PROPERTY superclass"
   (is-a PHYSICAL-PROPERTY LEVEL-3-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot family
      (default ENVIRONMENTAL))
```

```
)   ;end defclass ENVIRONMENTAL-PROPERTY

(defclass PROCESS-ECONOMICS
   "PROCESS-ECONOMICS is a THIRD-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-PROPERTY superclass"
   (is-a PROCESS-PROPERTY LEVEL-3-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot family
      (default ECONOMIC))
)   ;end defclass PROCESS-ECONOMICS

(defclass PROCESS-PARAMETER
   "PROCESS-PARAMETER is a THIRD-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-PROPERTY superclass"
   (is-a PROCESS-PROPERTY LEVEL-3-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot family
      (default PARAMETER))
)   ;end defclass PROCESS-PARAMETER

(defclass PROCESS-METHOD
   "PROCESS-METHOD is a THIRD-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-PROPERTY superclass"
   (is-a PROCESS-PROPERTY LEVEL-3-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot family
      (default METHOD))
)   ;end defclass PROCESS-METHOD

(defclass PROCESS-LIMITS
   "PROCESS-LIMITS is a THIRD-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-PROPERTY superclass"
   (is-a PROCESS-PROPERTY LEVEL-3-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot family
      (default LIMITS))
)   ;end defclass PROCESS-LIMITS

(defclass THERMAL-CONDUCTIVITY
   "THERMAL-CONDUCTIVITY is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the THERMAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY THERMAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default HEAT-TRANSFER))
   (slot units
      (default Watt/meter-deg-K))
   (slot symbol
      (default K))
```

```
)   ;end defclass THERMAL-CONDUCTIVITY

(defclass SPECIFIC-HEAT
   "SPECIFIC-HEAT is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the THERMAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY THERMAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default HEAT-TRANSFER))
   (slot units
      (default Joule/kilogram-deg-K))
   (slot symbol
      (default C))
)   ;end defclass SPECIFIC-HEAT

(defclass THERMAL-EXPANSION
   "THERMAL-EXPANSION is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the THERMAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY THERMAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default HEAT-TRANSFER))
   (slot units
      (default meter/meter-deg-K))
   (slot symbol
      (default ALPHA))
)   ;end defclass THERMAL-CONDUCTIVITY

(defclass THERMAL-DIFFUSIVITY
   "THERMAL-DIFFUSIVITY is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the THERMAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY THERMAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default HEAT-TRANSFER))
   (slot units
      (default meter*meter/second))
   (slot symbol
      (default ALPHA-sub-diff))
)   ;end defclass THERMAL-DIFFUSIVITY

(defclass THERMAL-SHOCK
   "THERMAL-SHOCK is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the THERMAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY THERMAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default HEAT-TRANSFER))
   (slot units
      (default Watt/meter-second))
```

```
   (slot symbol
      (default Th-sub-shock))
)   ;end defclass THERMAL-SHOCK

(defclass USAGE-TEMPERATURE
   "USAGE-TEMPERATURE is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the THERMAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY THERMAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default HEAT-TRANSFER))
   (slot units
      (default deg-K))
   (slot symbol
      (default T-sub-max))
)   ;end defclass USAGE-TEMPERATURE

(defclass MELT-POINT
   "MELT-POINT is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the THERMAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY THERMAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default HEAT-TRANSFER))
   (slot units
      (default deg-K))
   (slot symbol
      (default T-sub-melt))
)   ;end defclass MELT-POINT

(defclass YOUNGS-MODULUS
   "YOUNGS-MODULUS is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MECHANICAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY MECHANICAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default STRESS-STRAIN))
   (slot units
      (default Newton/meter*meter))
   (slot symbol
      (default E))
)   ;end defclass YOUNGS-MODULUS

(defclass SHEAR-MODULUS
   "SHEAR-MODULUS is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MECHANICAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY MECHANICAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default STRESS-STRAIN))
```

```
   (slot units
      (default Newton/meter*meter))
   (slot symbol
      (default G))
)   ;end defclass SHEAR-MODULUS

(defclass POISSON-RATIO
   "POISSON-RATIO is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MECHANICAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY MECHANICAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default STRESS-STRAIN))
   (slot symbol
      (default NU))
)   ;end defclass POISSON-RATIO

(defclass DENSITY
   "DENSITY is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MECHANICAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY MECHANICAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default MASS-VOLUME))
   (slot units
      (default kilogram/meter*meter*meter))
   (slot symbol
      (default RHO))
)   ;end defclass DENSITY

(defclass FRICTION-COEFFICIENT-DRY
   "FRICTION-COEFFICIENT-DRY is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MECHANICAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY MECHANICAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default FRICTION))
   (slot symbol
      (default MU-sub-dry))
)   ;end defclass FRICTION-COEFFICIENT-DRY

(defclass FRICTION-COEFFICIENT-LUB
   "FRICTION-COEFFICIENT-LUB is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MECHANICAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY MECHANICAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default FRICTION))
   (slot symbol
      (default MU-sub-lub))
```

```
)   ;end defclass FRICTION-COEFFICIENT-LUB

(defclass ROUGHNESS
   "ROUGHNESS is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MECHANICAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY MECHANICAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default FRICTION))
   (slot units
      (default meter))
   (slot symbol
      (default R))
)   ;end defclass ROUGHNESS

(defclass HARDNESS
   "HARDNESS is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MECHANICAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY MECHANICAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default STRESS-STRAIN))
   (slot units
      (default kilogram/millimeter*millimeter))   ;Knoop hardness
   (slot symbol
      (default H-sub-K))
)   ;end defclass HARDNESS

(defclass TOUGHNESS
   "TOUGHNESS is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MECHANICAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY MECHANICAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default STRESS-STRAIN))
   (slot units
      (default Newton/meter*meter))
   (slot symbol
      (default TNS))
)   ;end defclass TOUGHNESS

(defclass CRACK-RESISTANCE
   "CRACK-RESISTANCE is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the MECHANICAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY MECHANICAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default FRACTURE))
   (slot units
      (default Newton/meter*meter))   ;strain energy?
```

```
   (slot symbol
      (default CR))
)   ;end defclass CRACK-RESISTANCE

(defclass LUB-FILM-THKNS
   "LUB-FILM-THKNS is a FOURTH-level attribute specialization of the
   TECHNOLOGY attribute hierarchy under the ENVIRONMENTAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY ENVIRONMENTAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default HYDRODYNAMICS))
   (slot units
      (default meter))
   (slot symbol
      (default h-sub-lub))
)   ;end defclass LUB-FILM-THKNS

(defclass LUB-COMPATIBILITY
   "LUB-COMPATIBILITY is a FOURTH-level attribute specialization of the
   TECHNOLOGY attribute hierarchy under the ENVIRONMENTAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY ENVIRONMENTAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default HYDRODYNAMICS))
   (multislot lub-list
      (create-accessor read-write)
      (access read-write))
)   ;end defclass LUB-COMPATIBILITY

(defclass OXIDATION-RESISTANCE
   "OXIDATION-RESISTANCE is a FOURTH-level attribute specialization of the
   TECHNOLOGY attribute hierarchy under the ENVIRONMENTAL-PROPERTY superclass"
   (is-a TECHNOLOGY-PROPERTY ENVIRONMENTAL-PROPERTY LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default OXIDATION))
)   ;end defclass OXIDATION-RESISTANCE

(defclass MATERIAL-COST
   "MATERIAL-COST is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-ECONOMICS superclass"
   (is-a TECHNOLOGY-PROPERTY PROCESS-ECONOMICS LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default COST-PER-GRAM))
   (slot units
      (default dollars/gram))
   (slot symbol
      (default C-sub-M))
)   ;end defclass MATERIAL-COST
```

```
(defclass CAPITAL-EQUIPMENT-COST
   "CAPITAL-EQUIPMENT-COST is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-ECONOMICS superclass"
   (is-a TECHNOLOGY-PROPERTY PROCESS-ECONOMICS LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default COST-PER-GRAM))
   (slot units
      (default dollars/gram))
   (slot symbol
      (default C-sub-CE))
)   ;end defclass CAPITAL-EQUIPMENT-COST

(defclass LABOR-COST
   "LABOR-COST is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-ECONOMICS superclass"
   (is-a TECHNOLOGY-PROPERTY PROCESS-ECONOMICS LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default COST-PER-GRAM))
   (slot units
      (default dollars/gram))
   (slot symbol
      (default C-sub-L))
)   ;end defclass LABOR-COST

(defclass POWER-COST
   "POWER-COST is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-ECONOMICS superclass"
   (is-a TECHNOLOGY-PROPERTY PROCESS-ECONOMICS LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default COST-PER-GRAM))
   (slot units
      (default dollars/gram))
   (slot symbol
      (default C-sub-P))
)   ;end defclass POWER-COST

(defclass OVERHEAD-COST
   "OVERHEAD-COST is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-ECONOMICS superclass"
   (is-a TECHNOLOGY-PROPERTY PROCESS-ECONOMICS LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default COST-PER-GRAM))
   (slot units
      (default dollars/gram))
   (slot symbol
```

```
      (default C-sub-O))
)   ;end defclass OVERHEAD-COST

(defclass PROCESS-PRESSURE
   "PROCESS-PRESSURE is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-PARAMETER superclass"
   (is-a TECHNOLOGY-PROPERTY PROCESS-PARAMETER LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default PRESSURE))
   (slot units
      (default Newton/meter*meter))
   (slot symbol
      (default P-sub-p))
)   ;end defclass PROCESS-PRESSURE

(defclass PROCESS-ATMOSPHERE
   "PROCESS-ATMOSPHERE is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-PARAMETER superclass"
   (is-a TECHNOLOGY-PROPERTY PROCESS-PARAMETER LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default ATMOSPHERE))
   (multislot atmosphere-list
      (create-accessor read-write)
      (access read-write)
      (type LEXEME)
      (default ?DERIVE))
)   ;end defclass PROCESS-ATMOSPHERE

(defclass PROCESS-SURFACE-TEMPERATURE
   "PROCESS-SURFACE-TEMPERATURE is a FOURTH-level attribute specialization of
the
    TECHNOLOGY attribute hierarchy under the PROCESS-PARAMETER superclass"
   (is-a TECHNOLOGY-PROPERTY PROCESS-PARAMETER LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default TEMPERATURE))
   (slot units
      (default deg-C))
   (slot symbol
      (default T-sub-surf))
)   ;end defclass PROCESS-SURFACE-TEMPERATURE

(defclass METHOD-PREPARATION
   "METHOD-PREPARATION is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-METHOD superclass"
   (is-a MFG-PROCESS-METHOD-LIST PROCESS-METHOD)
   (role concrete)
   (pattern-match reactive)
   (slot species
```

```
      (default PREPARATION))
)   ;end defclass METHOD-PREPARATION

(defclass METHOD-APPLICATION
   "METHOD-APPLICATION is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-METHOD superclass"
   (is-a MFG-PROCESS-METHOD-LIST PROCESS-METHOD)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default APPLICATION))
)   ;end defclass METHOD-APPLICATION

(defclass METHOD-FINISHING
   "METHOD-FINISHING is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-METHOD superclass"
   (is-a MFG-PROCESS-METHOD-LIST PROCESS-METHOD)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default FINISHING))
)  ;end defclass METHOD-FINISHING

(defclass PROCESS-LIMITS-THKNS
   "PROCESS-LIMITS-THKNS is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-LIMITS superclass"
   (is-a TECHNOLOGY-PROPERTY PROCESS-LIMITS LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default LENGTH))
   (slot units
      (default meter))
   (slot symbol
      (default h-sub-0))
   (slot maximum
      (create-accessor read-write)
      (access read-write))
   (slot minimum
      (create-accessor read-write)
      (access read-write))
)   ;end defclass PROCESS-LIMITS-THKNS

(defclass PROCESS-LIMITS-SPATIAL
   "PROCESS-LIMITS-SPATIAL is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-LIMITS superclass"
   (is-a TECHNOLOGY-PROPERTY PROCESS-LIMITS LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default VOLUME))
   (slot units
      (default meter*meter*meter))
   (slot symbol
```

```
      (default V-sub-sp))
)   ;end defclass PROCESS-LIMITS-THKNS

(defclass PROCESS-LIMITS-DEPOSITION
   "PROCESS-LIMITS-DEPOSITION is a FOURTH-level attribute specialization of the
    TECHNOLOGY attribute hierarchy under the PROCESS-LIMITS superclass"
   (is-a TECHNOLOGY-PROPERTY PROCESS-LIMITS LEVEL-4-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot species
      (default DEPOSITION-RATE))
   (slot units
      (default gram/hour))
   (slot symbol
      (default M-sub-t))
)   ;end defclass PROCESS-LIMITS-DEPOSITION
```

Governing Equations.CLP

```
(defclass GOVERNING-EQUATION
   "GOVERNING-EQUATION is a first-level attribute specialization of
    the TECHNOLOGY attribute hierarchy"
   (is-a TECHNOLOGY LEVEL-1-ATTRIBUTE)
   (role abstract)
   (pattern-match non-reactive)
   (slot classification
      (default EQUATION))
)   ;end defclass GOVERNING-EQUATION

(defclass EMPIRICAL-RELATIONSHIP
   "EMPIRICAL-RELATIONSHIP is a first-level attribute specialization
    of the GOVERNING-EQUATION class within the TECHNOLOGY attribute
    hierarchy"
   (is-a GOVERNING-EQUATION)
   (role concrete)
   (pattern-match reactive)
   (slot relationship-name
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type LEXEME)
      (default ?NONE))       ; must specify for each instance
   (slot c-header            ; function prototype for c program
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type STRING)
      (default ?NONE))       ; must specify for each instance
) ; end defclass EMPIRICAL-RELATIONSHIP
```

```
(defclass OBJECTIVE-FUNCTION
   "OBJECTIVE-FUNCTION is a first-level attribute specialization
    of the GOVERNING-EQUATION class within the TECHNOLOGY attribute
    hierarchy. This function will be used as the objective to minimize
    in the optimization phase."
   (is-a GOVERNING-EQUATION)
   (role concrete)
   (pattern-match reactive)
   (slot function-name
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type LEXEME)
      (default ?NONE))      ; must specify for each instance
   (slot c-header           ; function prototype for c program
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type STRING)
      (default ?NONE))      ; must specify for each instance
) ; end defclass OBJECTIVE-FUNCTION

(defclass DESIGN-PARAMETER
   "DESIGN-PARAMETER is a first-level attribute specialization
    of the GOVERNING-EQUATION class within the TECHNOLOGY attribute
    hierarchy"
   (is-a GOVERNING-EQUATION)
   (role concrete)
   (pattern-match reactive)
   (slot parameter-name
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type LEXEME)
      (default ?NONE))      ; must specify for each instance
   (slot c-declaration      ; variable declaration for c program
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type STRING)
      (default ?NONE))      ; must specify for each instance
   (slot c-type             ; variable type declaration for c program
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type STRING)
      (allowed-strings
                 "int"
                 "short"
                 "long"
                 "float"
                 "double"
                 "long double")
```

```
      (default ?NONE))       ; must specify for each instance
) ; end defclass DESIGN-PARAMETER

(defclass CONSTANT
   "CONSTANT is a first-level attribute specialization
    of the GOVERNING-EQUATION class within the TECHNOLOGY attribute
    hierarchy"
   (is-a GOVERNING-EQUATION)
   (role concrete)
   (pattern-match reactive)
   (slot constant-name
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type LEXEME)
      (default ?NONE))       ; must specify for each instance
   (slot c-define            ; #DEFINE for c program
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type STRING)
      (default ?NONE))       ; must specify for each instance
   (slot c-type              ; variable type declaration for c program
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type STRING)
      (allowed-strings
                     "int"
                     "short"
                     "long"
                     "float"
                     "double"
                     "long double")
      (default ?NONE))       ; must specify for each instance
) ; end defclass CONSTANT

(defmessage-handler EMPIRICAL-RELATIONSHIP print-header (?LogicalName)
   (format ?LogicalName "    %s%n" ?self:c-header)
); end defmessage-handler EMPIRICAL-RELATIONSHIP print-header

(defmessage-handler OBJECTIVE-FUNCTION print-header (?LogicalName)
   (format ?LogicalName "    %s%n" ?self:c-header)
); end defmessage-handler OBJECTIVE-FUNCTION print-header

(defmessage-handler DESIGN-PARAMETER print-declaration (?LogicalName)
   (format ?LogicalName "    %s%n" ?self:c-declaration)
); end defmessage-handler DESIGN-PARAMETER print-declaration

(defmessage-handler CONSTANT print-define (?LogicalName)
   (format ?LogicalName "    %s%n" ?self:c-define)
); end defmessage-handler CONSTANT print-define
```

qualitative needs.CLP

```
(defclass QUALITATIVE-REQTS
   "QUALITATIVE-REQTS is a first-level attribute specialization of the
    TECHNOLOGY (applications) attribute hierarchy"
   (is-a TECHNOLOGY LEVEL-1-ATTRIBUTE)
   (role concrete)
   (pattern-match reactive)
   (slot technology-type
      (default APPLICATION))
   (slot classification
      (default QUALITATIVE-REQTS))
   (slot criteria-category      ; i.e. surface-contact
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type LEXEME)
      (default ?NONE))       ; must specify for each instance
   (slot criteria-name      ; i.e. slide-to-roll-ratio, contact-stress
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type LEXEME)
      (default ?NONE))       ; must specify for each instance
   (slot qualitative-value
      (create-accessor read-write)
      (access read-write)
      (type LEXEME)
      (allowed-values
                     VERY-LOW
                     LOW
                     MEDIUM-LOW
                     MEDIUM
                     MEDIUM-HIGH
                     HIGH
                     VERY-HIGH)
      (default ?NONE))       ; must specify for each instance
)   ;end defclass QUALITATIVE-REQTS
```

Match Table.CLP

```
(defclass MATCH-TABLE
   "MATCH-TABLE is a tabular structure to hold instances where the
    APPLICATION needs match DISCOVERY capabilities."
   (is-a USER)
   (role concrete)
```

```
(pattern-match reactive)
(slot discovery-name
   (create-accessor read-write)
   (access initialize-only)
   (storage local)
   (type LEXEME)
   (default ?NONE))
(slot application-name
   (create-accessor read-write)
   (access initialize-only)
   (storage local)
   (type LEXEME)
   (default ?NONE))
(slot criterion-matched
   (create-accessor read-write)
   (access initialize-only)
   (storage local)
   (type LEXEME)
   (default ?NONE))
(slot discovery-qualitative-value
   (create-accessor read-write)
   (access initialize-only)
   (storage local)
   (type LEXEME)
   (default ?NONE))
(slot application-qualitative-value
   (create-accessor read-write)
   (access initialize-only)
   (storage local)
   (type LEXEME)
   (default ?NONE))
(slot match-goodness
   (create-accessor read-write)
   (access initialize-only)
   (storage local)
   (type LEXEME)
   (default ?NONE))
(slot match-id
   (create-accessor read-write)
   (access initialize-only)
   (storage local)
   (type SYMBOL)
   (default-dynamic (gensym*)))
(slot discovery-attribute-address
   (create-accessor read-write)
   (access initialize-only)
   (storage local)
   (type INSTANCE)
   (default ?NONE))
(slot application-attribute-address
   (create-accessor read-write)
   (access initialize-only)
   (storage local)
   (type INSTANCE)
```

```
      (default ?NONE))
   (slot tally-update
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type SYMBOL)
      (allowed-symbols TRUE FALSE)
      (default FALSE))
); end defclass MATCH-TABLE
(defmessage-handler MATCH-TABLE print-MATCH-TABLE (?LogicalName)
   (format ?LogicalName "%n")
   (format ?LogicalName "+-----------------------------------------------------
------------------+%n")
   (format ?LogicalName "|                           Match
Table                          |%n")
   (format ?LogicalName "| Criterion Matched:        %20s
|%n"
      ?self:criterion-matched)
   (format ?LogicalName "| Match Goodness:           %20s
|%n"
      ?self:match-goodness)
   (format ?LogicalName "| Match Instance Identifier: %20s
|%n"
      (instance-name ?self))
   (format ?LogicalName "| Match Identifier:         %20s
|%n"
      ?self:match-id)
   (format ?LogicalName "+-----------------------------------------------------
------------------+%n")
   (format ?LogicalName "| Category                  Discovery
Application          |%n")
   (format ?LogicalName "+-----------------------------------------------------
------------------+%n")
   (format ?LogicalName "| Name:                %20s    %20s      |%n"
      ?self:discovery-name ?self:application-name)
   (format ?LogicalName "| Qualitative Value: %20s    %20s      |%n"
      ?self:discovery-qualitative-value ?self:application-qualitative-value)
   (format ?LogicalName "| Attribute Address: %20s    %20s      |%n"
      (instance-name-to-symbol (instance-name ?self:discovery-attribute-
address))
      (instance-name-to-symbol (instance-name ?self:application-attribute-
address)))
   (format ?LogicalName "+-----------------------------------------------------
------------------+%n")
   (format ?LogicalName "%n")
); end defmessage-handler MATCH-TABLE print-MATCH-TABLE
```

Match Tally Class.CLP

```
(defclass MATCH-TALLY
```

```
   "The MATCH-TALLY class holds a running total for DISCOVERY-
    APPLICATION matched pairs. MATCH-TALLY tracks the names of
    attributes that match and the total number of STRONG and WEAK
    matches."
   (is-a USER)
   (role concrete)
   (pattern-match reactive)
   (slot discovery-name
      (create-accessor read-write)
      (access initialize-only)
      (storage local)
      (type LEXEME)
      (default ?NONE))
   (slot application-name
      (create-accessor read-write)
      (access initialize-only)
      (storage local)
      (type LEXEME)
      (default ?NONE))
   (multislot strong-matches
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type LEXEME)
      (default ?DERIVE))
   (multislot weak-matches
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type LEXEME)
      (default ?DERIVE))
   (slot strong-count
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type INTEGER)
      (default 0))
   (slot weak-count
      (create-accessor read-write)
      (access read-write)
      (storage local)
      (type INTEGER)
      (default 0))
); end deftemplate MATCH-TALLY

(defmessage-handler MATCH-TALLY add-to-tally (?criterion ?fit)
   "MATCH-TALLY add-to-tally updates the MATCH-TALLY for a
    DISCOVERY-APPLICATION match. The ?fit determines whether the
    ?criterion matched is STRONG or WEAK. The slots weak-count
    and strong-count get incremented accordingly. The multifield slots
    for strong-matches and weak-matches get updated with the new
    ?criterion."
   (switch ?fit
      (case STRONG then
```

```
        (bind ?self:strong-count (+ 1 ?self:strong-count))
        (bind ?self:strong-matches ?self:strong-matches ?criterion)
     ); end case STRONG
     (case WEAK then
        (bind ?self:weak-count (+ 1 ?self:weak-count))
        (bind ?self:weak-matches ?self:weak-matches ?criterion)
     ); end case WEAK
   ); end switch
); end defmessage-handler MATCH-TALLY add-to-tally

(defmessage-handler MATCH-TALLY print-statistics (?LogicalName)
   "The defmessage-handler MATCH-TALLY print-statistics lists
    strong and weak criteria matches and provides a numerical
    score for the success of the match"
   (format ?LogicalName "%n%n")
   (format ?LogicalName "+-----------------------------------------------------
+%n")
   (format ?LogicalName "|                  Match Statistics
|%n")
   (format ?LogicalName "|                                                     |%n")
   (format ?LogicalName "| Discovery:        %20s                 |%n"
      ?self:discovery-name)
   (format ?LogicalName "| Application:      %20s                 |%n"
      ?self:application-name)
   (format ?LogicalName "+-----------------------------------------------------
+%n")
   (format ?LogicalName "| Strong Matches:
|%n")
   (bind ?n 1)
   (loop-for-count ?self:strong-count
   (format ?LogicalName "|     %20s                          |%n" (nth$ ?n
?self:strong-matches))
   (bind ?n (+ ?n 1)))

   (format ?LogicalName "+-----------------------------------------------------
+%n")
   (format ?LogicalName "| Weak Matches:
|%n")
   (bind ?n 1)
   (loop-for-count ?self:weak-count
   (format ?LogicalName "|     %20s                          |%n" (nth$ ?n
?self:weak-matches))
   (bind ?n (+ ?n 1)))

   (format ?LogicalName "+-----------------------------------------------------
+%n")
   (format ?LogicalName "| Score = %4.2f
|%n"
      (/ ?self:strong-count (+ ?self:strong-count ?self:weak-count)))
   (format ?LogicalName "+-----------------------------------------------------
+%n")
   (format ?LogicalName "%n%n")
); end defmessage-handler MATCH-TALLY print-statistics
```

B.2.2 Template Definitions

Match Templates.CLP

```
(deftemplate MATCH-TALLY
   "The MATCH-TALLY template holds a running total for DISCOVERY-
    APPLICATION matched pairs. MATCH-TALLY tracks the names of
    attributes that match and the total number of STRONG and WEAK
    matches."
   (slot discovery-name
      (type SYMBOL)
      (default ?DERIVE))
   (slot application-name
      (type SYMBOL)
      (default ?DERIVE))
   (multislot strong-matches
      (type SYMBOL))
   (multislot weak-matches
      (type SYMBOL))
   (slot strong-count
      (type INTEGER)
      (default 0))
   (slot weak-count
      (type INTEGER)
      (default 0))
); end deftemplate MATCH-TALLY

(deftemplate MATCH-STATUS
   "MATCH-STATUS indicates whether the MATCH-TALLY has been initialized
    for a given DISCOVERY-APPLICATION pair. This fact template prevents
    duplicate MATCH-TALLY template instances from being created."
   (slot discovery-name
      (type SYMBOL)
      (default ?DERIVE))
   (slot application-name
      (type SYMBOL)
      (default ?DERIVE))
   (slot status
      (type SYMBOL)
      (allowed-symbols NOT-READY READY)
      (default READY))
); end deftemplate MATCH-STATUS
```

B.2.3 Rule Definitions

Qualitative Rules.CLP

```
(defrule DECOMPOSE-CONTACT-STRESS
   "Rule DECOMPOSE-CONTACT-STRESS matches instances of the class
    QUALITATIVE-REQTS where:
      criteria-category    =    SURFACE-CONTACT
      criteria-name        =    SURFACE-STRESS
      criteria-name        =    SLIDE-TO-ROLL-RATIO
    Instances of the following classes are created for use in the
    DISCOVERY -- APPLICATION matching process:
      THERMAL-CONDUCTIVITY
      SPECIFIC-HEAT
      DENSITY"
   (object (is-a QUALITATIVE-REQTS)
           (criteria-category SURFACE-CONTACT)
           (criteria-name CONTACT-STRESS)
           (technology-name ?tname)
           (qualitative-value ?qvalue&HIGH | MEDIUM-HIGH | VERY-HIGH)
           (technology-type APPLICATION)
           (classification QUALITATIVE-REQTS))
   (object (is-a QUALITATIVE-REQTS)
           (criteria-name SLIDE-TO-ROLL-RATIO)
           (technology-name ?tname)
           (qualitative-value ?qvalue&HIGH | MEDIUM-HIGH | VERY-HIGH))
=>
   (make-instance (+ ?tname -K) of THERMAL-CONDUCTIVITY
      (qualitative-value ?qvalue)
      (technology-type APPLICATION)
      (technology-name ?tname)
      (phylum SURFACE-TREATMENT)
   )   ; end make-instance ?tname-K

   (make-instance (+ ?tname -C) of SPECIFIC-HEAT
      (qualitative-value MEDIUM)
      (technology-type APPLICATION)
      (technology-name ?tname)
      (phylum SURFACE-TREATMENT)
   )   ; end instance ?tname-C

   (make-instance (+ ?tname -RHO) of DENSITY
      (qualitative-value MEDIUM-LOW)
      (technology-type APPLICATION)
      (technology-name ?tname)
      (phylum SURFACE-TREATMENT)
   )   ; end instance ?tname-RHO

)   ; end defrule DECOMPOSE-CONTACT-STRESS
```

Matching Rules.CLP

```
(defrule MATCH-REQT-CAPABILITY
   "This rule builds a table of instances for DISCOVERY capabilities that
    match APPLICATION requirements. The matching table has the following
    attributes for the prelimnary ranking process:
      discovery-name
      application-name
      criterion-matched
      discovery-qualitative-value
      application-qualitative-value
      match-goodness
      match-id
      discovery-attribute-address
      application-attribute-address
   "
   ?appl <- (object (is-a ?x)
                    (species ?spec)
                    (technology-type APPLICATION))
   ?disc <- (object (is-a ?x)
                    (species ?spec)
                    (technology-type DISCOVERY))
=>
   (make-instance of MATCH-TABLE
      (discovery-attribute-address ?disc)
      (application-attribute-address ?appl)
      (discovery-name (send ?disc get-technology-name))
      (application-name (send ?appl get-technology-name))
      (criterion-matched (class ?appl))
      (discovery-qualitative-value (send ?disc get-qualitative-value))
      (application-qualitative-value (send ?appl get-qualitative-value))
      (match-goodness (match-rating (send ?appl get-qualitative-value)
                                    (send ?disc get-qualitative-value)
                                    (send ?appl get-assessment-criteria)))
   )   ;end make-instance MATCH-TABLE
)   ;end defrule MATCH-REQT-CAPABILITY

;(defrule INTIALIZE-MATCH-TALLY
;   "INTIALIZE-MATCH-TALLY asserts a fact template to track DISCOVERY-
;    APPLICATION matches. Attributes matched and the number of STRONG
;    and WEAK matches are also totaled."
;   (object (is-a MATCH-TABLE)
;           (discovery-name ?disc)
;           (application-name ?appl))
;   (not (MATCH-STATUS
;        (discovery-name ?disc)
;        (application-name ?appl)))
;=>
;   (assert (MATCH-TALLY
;           (discovery-name ?disc)
;           (application-name ?appl)))
```

```
;    (assert (MATCH-STATUS
;          (discovery-name ?disc)
;          (application-name ?appl)))
;); end defrule INTIALIZE-MATCH-TALLY
;
;(defrule UPDATE-MATCH-TALLY
;    "UPDATE-MATCH-TALLY updates the totals for the DISCOVERY-APPLICATION
;     match pairs. Attributes matched are added to a multislot field and
;     the numbers of STRONG and WEAK matches are totaled."
;    ?obj <- (object (is-a MATCH-TABLE)
;                    (discovery-name ?disc)
;                    (application-name ?appl)
;                    (criterion-matched ?crit)
;                    (match-goodness ?fit)
;                    (tally-update FALSE))
;    ?match <- (MATCH-TALLY
;                (discovery-name ?disc)
;                (application-name ?appl)
;                (strong-matches $?sm)
;                (weak-matches $?wm)
;                (strong-count ?sc)
;                (weak-count ?wc))
;=>
;    (if (str-compare ?fit STRONG)
;       then
;          (modify ?match
;                  (strong-matches ?sm ?crit)
;                  (strong-count (+ ?sc 1)))
;       else
;          (modify ?match
;                  (weak-matches ?wm ?crit)
;                  (weak-count (+ ?wc 1)))
;    ); end if
;    (send ?obj put-tally-update TRUE)
;); end defrule UPDATE-MATCH-TALLY

(defrule UPDATE-MATCH-TALLY-INSTANCE
    "UPDATE-MATCH-TALLY updates the totals for the DISCOVERY-APPLICATION
     match pairs. Attributes matched are added to a multislot field and
     the numbers of STRONG and WEAK matches are totaled."
    ?obj <- (object (is-a MATCH-TABLE)
                    (discovery-name ?disc)
                    (application-name ?appl)
                    (criterion-matched ?crit)
                    (match-goodness ?fit)
                    (tally-update FALSE))
=>
    (if (not
            (do-for-instance
               ((?mt MATCH-TALLY))
               (and (= ?mt:discovery-name ?disc)
                    (= ?mt:application-name ?appl))
               (send ?mt add-to-tally ?crit ?fit)
            ); end do-for-instance
```

```
        )
    then
      (switch ?fit
         (case STRONG then
            (make-instance of MATCH-TALLY
               (discovery-name ?disc)
               (application-name ?appl)
               (strong-matches ?crit)
               (strong-count 1)
            )
         ); end case STRONG
         (case WEAK then
            (make-instance of MATCH-TALLY
               (discovery-name ?disc)
               (application-name ?appl)
               (weak-matches ?crit)
               (weak-count 1)
            )
         ); end case WEAK
      ); end switch
   ); end if
   (send ?obj put-tally-update TRUE)
); end defrule UPDATE-MATCH-TALLY-INSTANCE
```

Menu Rules.CLP

```
(defrule SHOW-MAIN-MENU
   "SHOW-MAIN-MENU fires after all matching activity is completed. User
    can choose to display or save to files statistics relevant to the
    matching process. The user will also have the ability to download
    pertinent attribute information header files needed to run the
    analysis simulations."
   (declare (salience -10000))
=>
;   (CountTechMatches)
;   (CountAttrMatches)
   (bind ?nTmatches (CountTechMatches))
   (bind ?nAmatches (CountAttrMatches))
   (MainMenu 1 ?nTmatches ?nAmatches)
   (bind ?mainchoice (ValidUserInput (read t) (create$ 1 2 9)))
   (while (<> ?mainchoice 9)
      (switch ?mainchoice
         (case 1 then
            (DisplayMatchTable 1)
            (bind ?matchchoice (ValidUserInput (read t) (create$ 1 2 4 5 9)))
            (while (<> ?matchchoice 9)
               (switch ?matchchoice
                  (case 1 then
                     (PrintMatchTable t)
```

```
            (PressReturn))
         (case 2 then
            (GetAppendFile table)
            (PrintMatchTable table)
            (close table)
            (PressReturn))
         (case 4 then
            ;(printout t "match stats go here" crlf)
            (PrintMatchStats t)
            (PressReturn))
         (case 5 then
            (GetAppendFile stats)
            (PrintMatchStats stats)
            (close stats)
            (PressReturn))
      ); end switch
      (DisplayMatchTable 1)
     (bind ?matchchoice (ValidUserInput (read t) (create$ 1 2 4 5 9)))
   ); end while
); end case 1
(case 2 then
   (OutputForAnalysis 1)
   (bind ?outputchoice (ValidUserInput (read t) (create$ 1 2 3 4 5 6
9)))
   (while (<> ?outputchoice 9)
      (switch ?outputchoice
         (case 1 then
            (GetAppendFile er)
            (PrintEmpiricalRelationships APPLICATION er)
            (close er)
            (PressReturn))
         (case 2 then
            (GetAppendFile dp)
            (PrintDesignParameters APPLICATION dp)
            (close dp)
            (PressReturn))
         (case 3 then
            (GetAppendFile obf)
            (PrintObjectiveFunction APPLICATION obf)
            (close obf)
            (PressReturn))
         (case 4 then
            (GetAppendFile c)
            (PrintConstants APPLICATION c)
            (close c)
            (PressReturn))
         (case 5 then
            (GetAppendFile er)
            (PrintEmpiricalRelationships DISCOVERY er)
            (close er)
            (PressReturn))
         (case 6 then
            (GetAppendFile c)
            (PrintConstants DISCOVERY c)
```

```
                     (close c)
                     (PressReturn))
               ); end switch
               (OutputForAnalysis 1)
              (bind ?outputchoice (ValidUserInput (read t) (create$ 1 2 3 4 5
6 9)))
            );end while
         ); end case 2
      ); end switch
      (MainMenu 1 ?nTmatches ?nAmatches)
      (bind ?mainchoice (ValidUserInput (read t) (create$ 1 2 9)))
   ); end while stuff here...
   (printout t "we're all done now..." crlf)
); end defrule SHOW-MAIN-MENU
```

B.2.4 Functions

B.2.4.1 Generic methods

Generic Methods.CLP

```
(defgeneric + "overload to handle symbols and strings")
(defmethod + ((?a SYMBOL) (?b SYMBOL))     ;overload + to concatenate
   (sym-cat ?a ?b))                        ;SYMBOLS result is SYMBOL

(defmethod + ((?a STRING) (?b STRING))     ;overload + to concatenate
   (str-cat ?a ?b))                        ;STRINGS result is STRING

(defmethod + ((?a LEXEME) (?b LEXEME))     ;overload + to concatenate
   (sym-cat ?a ?b))                        ;LEXEMES result is SYMBOL

(defgeneric = "overload to handle symbols and strings")
(defmethod = ((?a SYMBOL) (?b SYMBOL))     ;overload = to compare
   (= (str-compare ?a ?b) 0))              ;SYMBOLS result is LOGICAL

(defmethod = ((?a STRING) (?b STRING))     ;overload = to compare
   (= (str-compare ?a ?b) 0))              ;STRINGS result is LOGICAL

(defmethod = ((?a LEXEME) (?b LEXEME))     ;overload = to compare
   (= (str-compare ?a ?b) 0))              ;LEXEMES result is LOGICAL
```

B.2.4.2 Deffunctions

Matching Functions.CLP

```
(deffunction match-rating (?appl-val ?disc-val ?assess-crit)
   (if (eq ?appl-val ?disc-val)
    then
      (return STRONG)
    else
      (switch ?appl-val
         (case VERY-LOW then
            (if (or (= ?disc-val LOW) (= ?disc-val MEDIUM-LOW))
             then
               (return STRONG)
             else
               (return WEAK)
           )    ;end if
         )    ;end case
         (case LOW then
            (if (or (= ?disc-val VERY-LOW) (= ?disc-val MEDIUM-LOW))
             then
               (return STRONG)
             else
               (return WEAK)
           )    ;end if
         )    ;end case
         (case MEDIUM-LOW then
            (if (or (= ?disc-val LOW) (= ?disc-val MEDIUM))
             then
               (return STRONG)
             else
               (return WEAK)
           )    ;end if
         )    ;end case
         (case MEDIUM then
            (if (or (= ?disc-val MEDIUM-LOW) (= ?disc-val MEDIUM-HIGH))
             then
               (return STRONG)
             else
               (return WEAK)
           )    ;end if
         )    ;end case
         (case MEDIUM-HIGH then
            (if (or (= ?disc-val MEDIUM) (= ?disc-val HIGH))
             then
               (return STRONG)
             else
               (return WEAK)
           )    ;end if
         )    ;end case
```

```
        (case HIGH then
           (if (or (= ?disc-val MEDIUM-HIGH) (= ?disc-val VERY-HIGH))
            then
              (return STRONG)
            else
              (return WEAK)
          )   ;end if
        )   ;end case
        (case VERY-HIGH then
           (if (or (= ?disc-val MEDIUM-HIGH) (= ?disc-val HIGH))
            then
              (return STRONG)
            else
              (return WEAK)
          )   ;end if
        )   ;end case
        (default WEAK)
     )   ;end switch
   )   ;end if
)   ;end deffunction match-rating
```

Menu System.CLP

```
(deffunction MainMenu (?menu ?matches ?attr-matches)
   (printout t "+----------------------------------------------------+" crlf)
   (printout t "|                                                    |" crlf)
   (printout t "|   Main Menu                                        |" crlf)
   (printout t "|                                                    |" crlf)
   (format   t "|   The Qualitative matching engine found %-02ld         |%n"
?matches)
   (format   t "|   Discovery--Application match(es) and %-02ld          |%n"
?attr-matches)
   (printout t "|   attribute capability--requirements matches.      |" crlf)
   (printout t "|                                                    |" crlf)
   (printout t "|   What would you like to do now? Please select     |" crlf)
   (printout t "|   one of the following options:                    |" crlf)
   (printout t "|                                                    |" crlf)
   (printout t "|      1   Display Match Table                       |" crlf)
   (printout t "|      2   Prepare matched technologies for          |" crlf)
   (printout t "|           quantitative analyses                    |" crlf)
   (printout t "|      9   Quit CLIPS                                |" crlf)
   (printout t "|                                                    |" crlf)
   (printout t "+----------------------------------------------------+" crlf)
   (return ?menu)
); end deffunction MainMenu


(deffunction DisplayMatchTable (?menu)
   (printout t "+----------------------------------------------------+" crlf)
```

```
   (printout t "|                                                      |" crlf)
   (printout t "|   Display Match Table Menu                           |" crlf)
   (printout t "|                                                      |" crlf)
   (printout t "|   What would you like to do now? Please select       |" crlf)
   (printout t "|   one of the following options:                      |" crlf)
   (printout t "|                                                      |" crlf)
   (printout t "|      1   Display Match Table                         |" crlf)
   (printout t "|      2   Save Match Table to an output file          |" crlf)
   (printout t "|                                                      |" crlf)
   (printout t "|      4   Display Match Statistics                    |" crlf)
   (printout t "|      5   Save Match Statistics to an output file     |" crlf)
   (printout t "|                                                      |" crlf)
   (printout t "|      9   Return to Main Menu                         |" crlf)
   (printout t "|                                                      |" crlf)
   (printout t "+------------------------------------------------------+" crlf)
   (return ?menu)
); end deffunction DisplayMatchTable

(deffunction OutputForAnalysis (?menu)
   (printout t "+------------------------------------------------------+" crlf)
   (printout t "|                                                      |" crlf)
   (printout t "|   Prepare Output Files for Analysis Menu             |" crlf)
   (printout t "|                                                      |" crlf)
   (printout t "|   What would you like to do now? Please select       |" crlf)
   (printout t "|   one of the following options:                      |" crlf)
   (printout t "|                                                      |" crlf)
   (printout t "|      1   Application governing equations             |" crlf)
   (printout t "|            c function prototypes                     |" crlf)
   (printout t "|      2   Application design parameter                |" crlf)
   (printout t "|            c prototypes                              |" crlf)
   (printout t "|      3   Application objective function              |" crlf)
   (printout t "|            c function prototypes                     |" crlf)
   (printout t "|      4   Application constants c prototypes          |" crlf)
   (printout t "|      5   Discovery governing equations               |" crlf)
   (printout t "|            c function prototypes                     |" crlf)
   (printout t "|      6   Discovery constants c prototypes            |" crlf)
   (printout t "|      9   Return to Main Menu                         |" crlf)
   (printout t "|                                                      |" crlf)
   (printout t "+------------------------------------------------------+" crlf)
   (return ?menu)
); end deffunction OutputForAnalysis

(deffunction ValidUserInput (?user $?allowed)
   (while (not (member$ ?user ?allowed))
      (format t "%s is not a valid choice. %n"
         (implode$ (create$ ?user)))
      (format t "Please enter from the following list: %n %s %n"
         (implode$ ?allowed))
      (bind ?user (read t)))
   (return ?user)
); end deffunction ValidUserInput

(deffunction PressReturn ()
   (format t "%n%n%n   Press <return> to continue...%n")
```

```
   (readline t)
); end deffunction PressReturn

(deffunction PrintMatchTable (?LogicalName)
   (do-for-all-instances
      ((?table MATCH-TABLE))
      TRUE
      (send ?table print-MATCH-TABLE ?LogicalName))
); end deffunction PrintMatchTable

(deffunction PrintMatchStats (?LogicalName)
   (do-for-all-instances
      ((?stats MATCH-TALLY))
      TRUE
      (send ?stats print-statistics ?LogicalName))
); end deffunction PrintMatchStats

(deffunction CountTechMatches ()
   (bind ?n 0)
   (do-for-all-instances
      ((?tally MATCH-TALLY))
      TRUE
      (bind ?n (+ ?n 1)))
   (return ?n)
); end of deffunction CountTechMatches

(deffunction CountAttrMatches ()
   (bind ?n 0)
   (do-for-all-instances
      ((?table MATCH-TABLE))
      TRUE
      (bind ?n (+ ?n 1)))
   (return ?n)
); end of deffunction CountAttrMatches

(deffunction PrintEmpiricalRelationships (?tech-type ?LogicalName)
   (format ?LogicalName "/* Begin %s Empirical Relationships */%n"
      ?tech-type)
   (do-for-all-instances
      ((?er EMPIRICAL-RELATIONSHIP))
      (eq ?tech-type ?er:technology-type)
      (send ?er print-header ?LogicalName))
   (format ?LogicalName "/* End %s Empirical Relationships */%n"
      ?tech-type)
); end deffunction PrintEmpiricalRelationships

(deffunction PrintObjectiveFunction (?tech-type ?LogicalName)
   (format ?LogicalName "/* Begin %s Objective Functions */%n"
      ?tech-type)
   (do-for-all-instances
      ((?of OBJECTIVE-FUNCTION))
      (eq ?tech-type ?of:technology-type)
      (send ?of print-header ?LogicalName))
   (format ?LogicalName "/* End %s Objective Functions */%n"
```

```
      ?tech-type)
); end deffunction PrintObjectiveFunction

(deffunction PrintDesignParameters (?tech-type ?LogicalName)
   (format ?LogicalName "/* Begin %s Design Parameters */%n"
      ?tech-type)
   (do-for-all-instances
      ((?dp DESIGN-PARAMETER))
      (eq ?tech-type ?dp:technology-type)
      (send ?dp print-declaration ?LogicalName))
   (format ?LogicalName "/* End %s Design Parameters */%n"
      ?tech-type)
); end deffunction PrintDesignParameters

(deffunction PrintConstants (?tech-type ?LogicalName)
   (format ?LogicalName "/* Begin %s Constants */%n"
      ?tech-type)
   (do-for-all-instances
      ((?c CONSTANT))
      (eq ?tech-type ?c:technology-type)
      (send ?c print-define ?LogicalName))
   (format ?LogicalName "/* End %s Constants */%n"
      ?tech-type)
); end deffunction PrintConstants
```

File Functions.CLP

```
(deffunction GetFile (?LogicalName ?FileMode)
   (format t "%n%nEnter output filename: %n")
   (bind ?filename (readline))
   (return (open ?filename ?LogicalName ?FileMode))
); end deffunction GetFile

(deffunction GetOutputFile (?LogicalName)
   (return (GetFile ?LogicalName "w"))
); end deffunction GetOutputFile

(deffunction GetInputFile (?LogicalName)
   (return (GetFile ?LogicalName "r"))
); end deffunction GetInputFile

(deffunction GetAppendFile (?LogicalName)
   (return (GetFile ?LogicalName "a"))
); end deffunction GetAppendFile

(deffunction GetReadAppendFile (?LogicalName)
   (return (GetFile ?LogicalName "r+"))
); end deffunction GetReadAppendFile
```

B.3 KNOWLEDGE BASE CONTENTS

B.3.1 DISCOVERY Instances

CVD-Diamond Instances.CLP

```
(definstances thermal-surface-properties
   (DIAMOND-CVD-K of THERMAL-CONDUCTIVITY
      (qualitative-value VERY-HIGH)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value 1540)
   )   ; end instance DIAMOND-CVD-K

   (DIAMOND-CVD-C of SPECIFIC-HEAT
      (qualitative-value MEDIUM)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value 853)
   )   ; end instance DIAMOND-CVD-C

   (DIAMOND-CVD-ALPHA of THERMAL-EXPANSION
      (qualitative-value VERY-LOW)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value 1.5E-6)
   )   ; end instance DIAMOND-CVD-ALPHA

   (DIAMOND-CVD-ALPHA-DIFF of THERMAL-DIFFUSIVITY
      (qualitative-value VERY-HIGH)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value 1291E-6)
   )   ; end instance DIAMOND-CVD-ALPHA-DIFF

   (DIAMOND-CVD-T-SHOCK of THERMAL-SHOCK
      (qualitative-value UNKNOWN)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value none)
   )   ; end instance DIAMOND-CVD-T-SHOCK

   (DIAMOND-CVD-USE-TEMP of USAGE-TEMPERATURE
      (qualitative-value VERY-HIGH)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value 2000)
   )   ; end instance DIAMOND-CVD-USE-TEMP
```

```
   (DIAMOND-CVD-MELT-POINT of MELT-POINT
      (qualitative-value VERY-HIGH)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value 3000)   ;DOESN'T MELT--VAPORIZES!
   )   ; end instance DIAMOND-CVD-MELT-POINT

)   ;end definstances thermal-surface-properties

(definstances mechanical-surface-properties
   (DIAMOND-CVD-E of YOUNGS-MODULUS
      (qualitative-value VERY-HIGH)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value 1.05E12)
   )   ; end instance DIAMOND-CVD-E

   (DIAMOND-CVD-G of SHEAR-MODULUS
      (qualitative-value VERY-HIGH)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value 5.5E11)
   )   ; end instance DIAMOND-CVD-G

   (DIAMOND-CVD-NU of POISSON-RATIO
      (qualitative-value VERY-LOW)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value 0.07)
   )   ; end instance DIAMOND-CVD-NU

   (DIAMOND-CVD-RHO of DENSITY
      (qualitative-value MEDIUM-LOW)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value 3500)
   )   ; end instance DIAMOND-CVD-RHO

   (DIAMOND-CVD-R of ROUGHNESS
      (qualitative-value MEDIUM-LOW)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value 0.1E-6)
   )   ; end instance DIAMOND-CVD-R

   (DIAMOND-CVD-HK of HARDNESS
      (qualitative-value VERY-HIGH)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (quantitative-value 5700)
   )   ; end instance DIAMOND-CVD-HK

)   ;end definstances mechanical-surface-properties
```

```
(definstances environmental-surface-properties
   (DIAMOND-CVD-OX-RESIST of OXIDATION-RESISTANCE
      (qualitative-value VERY-HIGH)
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
   )   ; end instance DIAMOND-CVD-OX-RESIST

   (DIAMOND-CVD-LUB-COMPAT of LUB-COMPATIBILITY
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (lub-list WATER OIL)
   )   ; end instance DIAMOND-CVD-LUB-COMPAT

)   ;end definstances environmental-surface-properties
```

CVD-Diamond Equations.CLP

```
(definstances diamond-cvd-equations
   "diamond-cvd-equations builds the equations necessary to design
    with the diamond CVD process."
   (DIAMOND-CVD-THETA-SSO of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-THETA-SSO)
      (c-header "float ThetaSSO( float PenDepth, float Length, \
       float LayerConductivity, float SubstrateConductivity, \
       float LayerThickness, float EntryLength );")
   ) ; end instance DIAMOND-CVD-THETA-SSO
   (DIAMOND-CVD-THETA-OS of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-THETA-OS)
      (c-header "float ThetaOS( float PenDepth, float Length, \
       float LayerConductivity, float SubstrateConductivity, \
       float LayerThickness, float EntryLength );")
   ) ; end instance DIAMOND-CVD-THETA-OS
   (DIAMOND-CVD-THETA-S of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-THETA-S)
      (c-header "float ThetaS( float PenDepth, float Length, \
       float LayerConductivity, float SubstrateConductivity );")
   ) ; end instance DIAMOND-CVD-THETA-S
   (DIAMOND-CVD-THETA-SO of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-THETA-SO)
      (c-header "float ThetaSO( float ThetaSSO, float ThetaS );")
   ) ; end instance DIAMOND-CVD-THETA-SO
```

```
(DIAMOND-CVD-THETA-O of EMPIRICAL-RELATIONSHIP
   (technology-name DIAMOND-CVD-COATING)
   (phylum SURFACE-TREATMENT)
   (relationship-name DIAMOND-CVD-THETA-O)
   (c-header "float ThetaO( float ThetaOS, float ThetaS );")
) ; end instance DIAMOND-CVD-THETA-O
(DIAMOND-CVD-T of EMPIRICAL-RELATIONSHIP
   (technology-name DIAMOND-CVD-COATING)
   (phylum SURFACE-TREATMENT)
   (relationship-name DIAMOND-CVD-TEMPERATURE)
   (c-header "float Temperature( float Theta, float HeatIn );")
) ; end instance DIAMOND-CVD-T
(DIAMOND-CVD-D-PEN of EMPIRICAL-RELATIONSHIP
   (technology-name DIAMOND-CVD-COATING)
   (phylum SURFACE-TREATMENT)
   (relationship-name DIAMOND-CVD-PEN-DEPTH)
   (c-header "float PenetrationDepth( float Conductivity, float Density, \
    float SpecificHeat, float Length, float Velocity );")
) ; end instance DIAMOND-CVD-D-PEN
(DIAMOND-CVD-L-ENTRY of EMPIRICAL-RELATIONSHIP
   (technology-name DIAMOND-CVD-COATING)
   (phylum SURFACE-TREATMENT)
   (relationship-name DIAMOND-CVD-ENTRY-LENGTH)
   (c-header "float EntryLength( float LayerConductivity, float
LayerDensity, \
    float LayerSpecificHeat, float Velocity, float LayerThickness );")
) ; end instance DIAMOND-CVD-L-ENTRY
(DIAMOND-CVD-SIGMA of EMPIRICAL-RELATIONSHIP
   (technology-name DIAMOND-CVD-COATING)
   (phylum SURFACE-TREATMENT)
   (relationship-name DIAMOND-CVD-NORMAL-STRESS)
   (c-header "float NormalStress( float YoungsModulus, float
ThermalExpansion, \
    float TemperatureRise );")
) ; end instance DIAMOND-CVD-SIGMA
(DIAMOND-CVD-TAU of EMPIRICAL-RELATIONSHIP
   (technology-name DIAMOND-CVD-COATING)
   (phylum SURFACE-TREATMENT)
   (relationship-name DIAMOND-CVD-SHEAR-STRESS)
   (c-header "float ShearStress( float LayerNormalStress, float
SubstrateNormalStress, \
    float LayerThickness, float LayerWidth );")
) ; end instance DIAMOND-CVD-TAU
(DIAMOND-CVD-THETA-SS of EMPIRICAL-RELATIONSHIP
   (technology-name DIAMOND-CVD-COATING)
   (phylum SURFACE-TREATMENT)
   (relationship-name DIAMOND-CVD-THETA-SS)
   (c-header "float ThetaSS( float ThetaInterface, float ThetaSO_IFSS, \
    float ThetaO_IFSS );")
) ; end instance DIAMOND-CVD-THETA-SS
(DIAMOND-CVD-DELTA of EMPIRICAL-RELATIONSHIP
   (technology-name DIAMOND-CVD-COATING)
   (phylum SURFACE-TREATMENT)
   (relationship-name DIAMOND-CVD-DELTA)
```

```
      (c-header "float Delta( float ThetaUncoatedSubstrate, float
ThetaSubstrate );")
   ) ; end instance DIAMOND-CVD-DELTA
   (DIAMOND-CVD-THETA-SS-P of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-THETA-SS-P)
      (c-header "float ThetaSS_P( float Delta, float ThetaSubstrate );")
   ) ; end instance DIAMOND-CVD-THETA-SS-P
   (DIAMOND-CVD-THETA-IF of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-THETA-IF)
      (c-header "float ThetaIF( float Delta, float Theta_IF );")
   ) ; end instance DIAMOND-CVD-THETA-IF
   (DIAMOND-CVD-THETA-D of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-THETA-D)
      (c-header "float ThetaD( float Delta, float Theta_D );")
   ) ; end instance DIAMOND-CVD-THETA-D
   (DIAMOND-CVD-TS of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-TEMPERATURE-UNCOATED)
     (c-header "float TemperatureS( float ThetaUncoatedSubstrate, float HeatIn
);")
   ) ; end instance DIAMOND-CVD-TS
   (DIAMOND-CVD-TSS of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-TEMPERATURE-SUBSTRATE)
     (c-header "float TemperatureSubstrate( float ThetaSubstrate, float HeatIn
);")
   ) ; end instance DIAMOND-CVD-TSS
   (DIAMOND-CVD-TIF of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-TEMPERATURE-INTERFACE)
      (c-header "float TemperatureInterface( float ThetaInterfaceLayer, float
HeatIn );")
   ) ; end instance DIAMOND-CVD-TIF
   (DIAMOND-CVD-TD of EMPIRICAL-RELATIONSHIP
      (technology-name DIAMOND-CVD-COATING)
      (phylum SURFACE-TREATMENT)
      (relationship-name DIAMOND-CVD-TEMPERATURE-DIAMOND)
      (c-header "float TemperatureDiamond( float ThetaDiamond, float HeatIn
);")
   ) ; end instance DIAMOND-CVD-TD
) ; end definstances diamond-cvd-equations
```

B.3.2 APPLICATION Instances

B.3.2.1 GEARS

Gear Instances.CLP

```
(definstances gear-application
   (GEAR-CONTACT-STRESS of QUALITATIVE-REQTS
      (technology-name GEARS)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (criteria-category SURFACE-CONTACT)
      (criteria-name CONTACT-STRESS)
      (qualitative-value HIGH)
   )   ; end instance GEAR-CONTACT-STRESS

   (GEAR-SLIDE-ROLL of QUALITATIVE-REQTS
      (technology-name GEARS)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (criteria-category SURFACE-CONTACT)
      (criteria-name SLIDE-TO-ROLL-RATIO)
      (qualitative-value HIGH)
   )   ; end instance GEAR-SLIDE-ROLL

); end definstances gear-application
```

Gear Equations.CLP

```
(definstances gear-equations
   "gear-equations builds the equations necessary to design gears."
   (GEAR-OBJ-FUNC of OBJECTIVE-FUNCTION
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (function-name GEAR-VOLUME)
      (c-header "float GearObjFunc( int GearRatio, float CenterDistance );")
   ) ; end instance GEAR-OBJ-FUNC
   (GEAR-XSTART of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-X-START)
      (c-header "float GearXStart( int GearRatio, int NPinion, float phi );")
   ) ; end instance GEAR-XSTART
   (GEAR-TPS of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
```

```
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-T-PINION-START)
      (c-header "float GearTPinionStart( float X, int GearRatio, \
       float phi, float psiP );")
   ) ; end instance GEAR-TPS
   (GEAR-TP of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-T-PINION)
      (c-header "float GearTPinion( float TPS, float f, float EEquiv, \
       float omegaPinion, float CenterDistance, float KrhoC );")
   ) ; end instance GEAR-TP
   (GEAR-A-WIDTH of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-CONTACT-WIDTH)
      (c-header "float GearContactWidth( float TangentLoad, float EEquiv, \
       float RadiusEquiv, float phi );")
   ) ; end instance GEAR-A-WIDTH
   (GEAR-Q-PINION of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-Q-PINION)
      (c-header "float GearQPinion( float KrhoC, float DeltaTPinion, \
       float aWidth, float VelocityPinion, float psiP );")
   ) ; end instance GEAR-Q-PINION
   (GEAR-COAT-AREA of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-COAT-AREA)
      (c-header "float GearCoatArea( float PitchDiameter, float FaceWidth );")
   ) ; end instance GEAR-COAT-AREA
   (GEAR-R-EFF of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-RADIUS-EFFECTIVE)
      (c-header "float GearEffectiveRadius( float CenterDistance, float X,
float phi );")
   ) ; end instance GEAR-R-EFF
   (GEAR-E-EFF of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-YOUNGS-MODULUS-EFF)
      (c-header "float GearEffectiveModulus( float EGear, float EPinion );")
   ) ; end instance GEAR-E-EFF
   (GEAR-PINION-PD of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
```

```
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-PINION-PITCH-DIA)
      (c-header "float GearPinionPitchDia( float CenterDistance, int GearRatio
);")
   ) ; end instance GEAR-PINION-PD
   (GEAR-V-PINION of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-PINION-VELOCITY)
      (c-header "float GearPinionVelocity( float omegaPinion, float
CenterDistance, \
       float X, float phi );")
   ) ; end instance GEAR-V-PINION
   (GEAR-V-GEAR of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-GEAR-VELOCITY)
      (c-header "float GearGearVelocity( int GearRatio, float omegaPinion, \
       float CenterDistance, float X, float phi );")
   ) ; end instance GEAR-V-GEAR
   (GEAR-V-SLIDE of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-SLIDE-VELOCITY)
      (c-header "float GearSlideVelocity( float Vpinion, float Vgear );")
   ) ; end instance GEAR-V-SLIDE
   (GEAR-V-ROLL of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-ROLL-VELOCITY)
      (c-header "float GearRollVelocity( float Vpinion, float Vgear );")
   ) ; end instance GEAR-V-ROLL
   (GEAR-SIGMA-H of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-CONTACT-STRESS)
      (c-header "float GearContactStress( float TangentLoad, float EEquiv, \
       float RadiusEquiv, float phi );")
   ) ; end instance GEAR-SIGMA-H
   (GEAR-SIGMA-B of EMPIRICAL-RELATIONSHIP
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-BENDING-STRESS)
      (c-header "float GearBendingStress( float TangentLoad, int NPinion, \
       float CenterDistance, int GearRatio );")
   ) ; end instance GEAR-SIGMA-B
   (GEAR-TORQUE-IN of EMPIRICAL-RELATIONSHIP
```

```
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (relationship-name GEAR-TORQUE-IN)
      (c-header "float GearTorque( float PitchDiameter, float TangentLoad );")
   ) ; end instance GEAR-SIGMA-B
) ; end definstances gear-equations

(definstances gear-design-parameters
   "gear-design-parameters defines the design decision parameters for gears"
   (GEAR-CD of DESIGN-PARAMETER
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (parameter-name GEAR-CENTER-DISTANCE)
      (c-declaration "float CenterDistance;")
      (c-type "float")
   ) ; end instance GEAR-CD
   (GEAR-NP of DESIGN-PARAMETER
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (parameter-name GEAR-N-PINION-TEETH)
      (c-declaration "int NPinion;")
      (c-type "int")
   ) ; end instance GEAR-NP
   (GEAR-MG of DESIGN-PARAMETER
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (parameter-name GEAR-GEAR-RATIO)
      (c-declaration "int GearRatio;")
      (c-type "int")
   ) ; end instance GEAR-MG
   (GEAR-OMEGA of DESIGN-PARAMETER
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (parameter-name GEAR-PINION-SPEED)
      (c-declaration "float omega;")
      (c-type "float")
   ) ; end instance GEAR-OMEGA
   (GEAR-TORQUE of DESIGN-PARAMETER
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (parameter-name GEAR-INPUT-TORQUE)
      (c-declaration "float TorqueIn;")
      (c-type "float")
   ) ; end instance GEAR-TORQUE
) ; end definstances gear-design-parameters

(definstances gear-constants
   "gear-constants defines the design constants for gears"
```

```
   (GEAR-T-P-MAX of CONSTANT
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (constant-name GEAR-TEMP-PINION-MAX-RISE)
      (c-define "#DEFINE GEAR-TEMP-PINION-MAX-RISE *")
      (c-type "float")
   ) ; end instance GEAR-T-P-MAX
   (GEAR-T-G-MAX of CONSTANT
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (constant-name GEAR-TEMP-GEAR-MAX-RISE)
      (c-define "#DEFINE GEAR-TEMP-GEAR-MAX-RISE *")
      (c-type "float")
   ) ; end instance GEAR-T-G-MAX
   (GEAR-SIGMA-B-MAX of CONSTANT
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (constant-name GEAR-MAX-BENDING-STRESS)
      (c-define "#DEFINE GEAR-MAX-BENDING-STRESS *")
      (c-type "float")
   ) ; end instance GEAR-SIGMA-B-MAX
   (GEAR-SIGMA-H-MAX of CONSTANT
      (technology-name GEARS)
      (technology-type APPLICATION)
      (phylum MECHANICAL-POWER-TRANSMISSION)
      (constant-name GEAR-MAX-CONTACT-STRESS)
      (c-define "#DEFINE GEAR-MAX-CONTACT-STRESS *")
      (c-type "float")
   ) ; end instance GEAR-SIGMA-H-MAX
) ; end definstances gear-constants
```

B.3.2.2 HI-LOAD-ROLLER-BRGS

Hi-Load-Roller-Brg Instances.CLP

```
(definstances brg-thermal-surface-properties
   (HI-LOAD-ROLLER-BRG-K of THERMAL-CONDUCTIVITY
      (qualitative-value MEDIUM)
      (technology-name HI-LOAD-ROLLER-BRG)
      (technology-type APPLICATION)
      (phylum ANTI-FRICTION-BRG)
      (assessment-criteria NEED)
   )   ; end instance HI-LOAD-ROLLER-BRG-K

   (HI-LOAD-ROLLER-BRG-C of SPECIFIC-HEAT
      (qualitative-value MEDIUM)
      (technology-name HI-LOAD-ROLLER-BRG)
```

```
      (technology-type APPLICATION)
      (phylum ANTI-FRICTION-BRG)
      (assessment-criteria NEED)
   )   ; end instance HI-LOAD-ROLLER-BRG-C

   (HI-LOAD-ROLLER-BRG-ALPHA of THERMAL-EXPANSION
      (qualitative-value MEDIUM)
      (technology-name HI-LOAD-ROLLER-BRG)
      (technology-type APPLICATION)
      (phylum ANTI-FRICTION-BRG)
      (assessment-criteria NEED)
   )   ; end instance HI-LOAD-ROLLER-BRG-ALPHA

   (HI-LOAD-ROLLER-BRG-USE-TEMP of USAGE-TEMPERATURE
      (qualitative-value LOW)
      (technology-name HI-LOAD-ROLLER-BRG)
      (technology-type APPLICATION)
      (phylum ANTI-FRICTION-BRG)
      (assessment-criteria NEED)
   )   ; end instance HI-LOAD-ROLLER-BRG-USE-TEMP

)   ;end definstances brg-thermal-surface-properties

(definstances brg-mechanical-surface-properties
   (HI-LOAD-ROLLER-BRG-E of YOUNGS-MODULUS
      (qualitative-value MEDIUM-HIGH)
      (technology-name HI-LOAD-ROLLER-BRG)
      (technology-type APPLICATION)
      (phylum ANTI-FRICTION-BRG)
      (assessment-criteria NEED)
   )   ; end instance HI-LOAD-ROLLER-BRG-E

   (HI-LOAD-ROLLER-BRG-G of SHEAR-MODULUS
      (qualitative-value MEDIUM-HIGH)
      (technology-name HI-LOAD-ROLLER-BRG)
      (technology-type APPLICATION)
      (phylum ANTI-FRICTION-BRG)
      (assessment-criteria NEED)
   )   ; end instance HI-LOAD-ROLLER-BRG-G

   (HI-LOAD-ROLLER-BRG-NU of POISSON-RATIO
      (qualitative-value MEDIUM)
      (technology-name HI-LOAD-ROLLER-BRG)
      (technology-type APPLICATION)
      (phylum ANTI-FRICTION-BRG)
      (assessment-criteria NEED)
   )   ; end instance HI-LOAD-ROLLER-BRG-NU

   (HI-LOAD-ROLLER-BRG-RHO of DENSITY
      (qualitative-value MEDIUM-LOW)
      (technology-name HI-LOAD-ROLLER-BRG)
      (technology-type APPLICATION)
      (phylum ANTI-FRICTION-BRG)
      (assessment-criteria NEED)
```

```
    )   ; end instance HI-LOAD-ROLLER-BRG-RHO

    (HI-LOAD-ROLLER-BRG-R of ROUGHNESS
       (qualitative-value LOW)
       (technology-name HI-LOAD-ROLLER-BRG)
       (technology-type APPLICATION)
       (phylum ANTI-FRICTION-BRG)
       (assessment-criteria NEED)
    )   ; end instance HI-LOAD-ROLLER-BRG-R

    (HI-LOAD-ROLLER-BRG-HK of HARDNESS
       (qualitative-value MEDIUM-HIGH)
       (technology-name HI-LOAD-ROLLER-BRG)
       (technology-type APPLICATION)
       (phylum ANTI-FRICTION-BRG)
       (assessment-criteria NEED)
    )   ; end instance HI-LOAD-ROLLER-BRG-HK

)   ;end definstances brg-mechanical-surface-properties

(definstances brg-environmental-surface-properties
    (HI-LOAD-ROLLER-BRG-OX-RESIST of OXIDATION-RESISTANCE
       (qualitative-value MEDIUM)
       (technology-name HI-LOAD-ROLLER-BRG)
       (assessment-criteria NEED)
       (technology-type APPLICATION)
       (phylum ANTI-FRICTION-BRG)
    )   ; end instance HI-LOAD-ROLLER-BRG-OX-RESIST

    (HI-LOAD-ROLLER-BRG-LUB-COMPAT of LUB-COMPATIBILITY
       (technology-name HI-LOAD-ROLLER-BRG)
       (technology-type APPLICATION)
       (phylum ANTI-FRICTION-BRG)
       (assessment-criteria NEED)
       (lub-list OIL)
    )   ; end instance HI-LOAD-ROLLER-BRG-LUB-COMPAT

)   ;end definstances brg-environmental-surface-properties
```

APPENDIX C
KNOWLEDGE BASE OUTPUT FILES

C.1 Matching Results

C.1.1 Match Table

match_table.out

```
+--------------------------------------------------------------------+
|                              Match Table                           |
| Criterion Matched:            LUB-COMPATIBILITY                    |
| Match Goodness:                          STRONG                    |
| Match Instance Identifier:                 gen1                    |
| Match Identifier:                          gen2                    |
+--------------------------------------------------------------------+
| Category                 Discovery               Application       |
+--------------------------------------------------------------------+
| Name:                DIAMOND-CVD-COATING     HI-LOAD-ROLLER-BRG    |
| Qualitative Value:              VERY-LOW               VERY-LOW    |
| Attribute Address: DIAMOND-CVD-LUB-COMPAT    HI-LOAD-ROLLER-BRG-LUB-
COMPAT      |
+--------------------------------------------------------------------+


+--------------------------------------------------------------------+
|                              Match Table                           |
| Criterion Matched:         OXIDATION-RESISTANCE                    |
| Match Goodness:                            WEAK                    |
| Match Instance Identifier:                 gen4                    |
| Match Identifier:                          gen5                    |
+--------------------------------------------------------------------+
| Category                 Discovery               Application       |
+--------------------------------------------------------------------+
| Name:                DIAMOND-CVD-COATING     HI-LOAD-ROLLER-BRG    |
| Qualitative Value:             VERY-HIGH                 MEDIUM    |
| Attribute Address: DIAMOND-CVD-OX-RESIST    HI-LOAD-ROLLER-BRG-OX-
RESIST      |
+--------------------------------------------------------------------+
```

```
+-------------------------------------------------------------------+
|                          Match Table                              |
| Criterion Matched:                    HARDNESS                    |
| Match Goodness:                           WEAK                    |
| Match Instance Identifier:                gen6                    |
| Match Identifier:                         gen7                    |
+-------------------------------------------------------------------+
| Category              Discovery              Application          |
+-------------------------------------------------------------------+
| Name:            DIAMOND-CVD-COATING     HI-LOAD-ROLLER-BRG       |
| Qualitative Value:          VERY-HIGH            MEDIUM-HIGH      |
| Attribute Address:     DIAMOND-CVD-HK   HI-LOAD-ROLLER-BRG-HK     |
+-------------------------------------------------------------------+


+-------------------------------------------------------------------+
|                          Match Table                              |
| Criterion Matched:                   ROUGHNESS                    |
| Match Goodness:                         STRONG                    |
| Match Instance Identifier:                gen8                    |
| Match Identifier:                         gen9                    |
+-------------------------------------------------------------------+
| Category              Discovery              Application          |
+-------------------------------------------------------------------+
| Name:            DIAMOND-CVD-COATING     HI-LOAD-ROLLER-BRG       |
| Qualitative Value:         MEDIUM-LOW                    LOW      |
| Attribute Address:      DIAMOND-CVD-R    HI-LOAD-ROLLER-BRG-R     |
+-------------------------------------------------------------------+


+-------------------------------------------------------------------+
|                          Match Table                              |
| Criterion Matched:                     DENSITY                    |
| Match Goodness:                         STRONG                    |
| Match Instance Identifier:               gen10                    |
| Match Identifier:                        gen11                    |
+-------------------------------------------------------------------+
| Category              Discovery              Application          |
+-------------------------------------------------------------------+
| Name:            DIAMOND-CVD-COATING     HI-LOAD-ROLLER-BRG       |
| Qualitative Value:         MEDIUM-LOW             MEDIUM-LOW      |
| Attribute Address:    DIAMOND-CVD-RHO   HI-LOAD-ROLLER-BRG-RHO
|
+-------------------------------------------------------------------+


+-------------------------------------------------------------------+
|                          Match Table                              |
| Criterion Matched:               POISSON-RATIO                    |
| Match Goodness:                           WEAK                    |
| Match Instance Identifier:               gen12                    |
| Match Identifier:                        gen13                    |
+-------------------------------------------------------------------+
```

```
| Category                 Discovery                Application        |
+----------------------------------------------------------------------+
| Name:              DIAMOND-CVD-COATING     HI-LOAD-ROLLER-BRG         |
| Qualitative Value:            VERY-LOW                 MEDIUM         |
| Attribute Address:      DIAMOND-CVD-NU   HI-LOAD-ROLLER-BRG-NU        |
+----------------------------------------------------------------------+


+----------------------------------------------------------------------+
|                              Match Table                             |
| Criterion Matched:                SHEAR-MODULUS                      |
| Match Goodness:                            WEAK                      |
| Match Instance Identifier:                gen14                      |
| Match Identifier:                         gen15                      |
+----------------------------------------------------------------------+
| Category                 Discovery                Application        |
+----------------------------------------------------------------------+
| Name:              DIAMOND-CVD-COATING     HI-LOAD-ROLLER-BRG         |
| Qualitative Value:           VERY-HIGH            MEDIUM-HIGH         |
| Attribute Address:       DIAMOND-CVD-G    HI-LOAD-ROLLER-BRG-G        |
+----------------------------------------------------------------------+


+----------------------------------------------------------------------+
|                              Match Table                             |
| Criterion Matched:               YOUNGS-MODULUS                      |
| Match Goodness:                            WEAK                      |
| Match Instance Identifier:                gen16                      |
| Match Identifier:                         gen17                      |
+----------------------------------------------------------------------+
| Category                 Discovery                Application        |
+----------------------------------------------------------------------+
| Name:              DIAMOND-CVD-COATING     HI-LOAD-ROLLER-BRG         |
| Qualitative Value:           VERY-HIGH            MEDIUM-HIGH         |
| Attribute Address:       DIAMOND-CVD-E    HI-LOAD-ROLLER-BRG-E        |
+----------------------------------------------------------------------+


+----------------------------------------------------------------------+
|                              Match Table                             |
| Criterion Matched:            USAGE-TEMPERATURE                      |
| Match Goodness:                            WEAK                      |
| Match Instance Identifier:                gen18                      |
| Match Identifier:                         gen19                      |
+----------------------------------------------------------------------+
| Category                 Discovery                Application        |
+----------------------------------------------------------------------+
| Name:              DIAMOND-CVD-COATING     HI-LOAD-ROLLER-BRG         |
| Qualitative Value:           VERY-HIGH                    LOW         |
| Attribute Address: DIAMOND-CVD-USE-TEMP  HI-LOAD-ROLLER-BRG-USE-TEMP
|
+----------------------------------------------------------------------+
```

```
+------------------------------------------------------------------------+
|                          Match Table                                   |
| Criterion Matched:       THERMAL-EXPANSION                             |
| Match Goodness:                       WEAK                             |
| Match Instance Identifier:           gen20                             |
| Match Identifier:                    gen21                             |
+------------------------------------------------------------------------+
| Category               Discovery              Application              |
+------------------------------------------------------------------------+
| Name:              DIAMOND-CVD-COATING     HI-LOAD-ROLLER-BRG          |
| Qualitative Value:            VERY-LOW                 MEDIUM          |
| Attribute Address:   DIAMOND-CVD-ALPHA   HI-LOAD-ROLLER-BRG-
ALPHA      |
+------------------------------------------------------------------------+


+------------------------------------------------------------------------+
|                          Match Table                                   |
| Criterion Matched:           SPECIFIC-HEAT                             |
| Match Goodness:                     STRONG                             |
| Match Instance Identifier:           gen22                             |
| Match Identifier:                    gen23                             |
+------------------------------------------------------------------------+
| Category               Discovery              Application              |
+------------------------------------------------------------------------+
| Name:              DIAMOND-CVD-COATING     HI-LOAD-ROLLER-BRG          |
| Qualitative Value:              MEDIUM                 MEDIUM          |
| Attribute Address:       DIAMOND-CVD-C   HI-LOAD-ROLLER-BRG-C          |
+------------------------------------------------------------------------+


+------------------------------------------------------------------------+
|                          Match Table                                   |
| Criterion Matched:    THERMAL-CONDUCTIVITY                             |
| Match Goodness:                       WEAK                             |
| Match Instance Identifier:           gen24                             |
| Match Identifier:                    gen25                             |
+------------------------------------------------------------------------+
| Category               Discovery              Application              |
+------------------------------------------------------------------------+
| Name:              DIAMOND-CVD-COATING     HI-LOAD-ROLLER-BRG          |
| Qualitative Value:           VERY-HIGH                 MEDIUM          |
| Attribute Address:       DIAMOND-CVD-K   HI-LOAD-ROLLER-BRG-K          |
+------------------------------------------------------------------------+


+------------------------------------------------------------------------+
|                          Match Table                                   |
| Criterion Matched:                 DENSITY                             |
| Match Goodness:                     STRONG                             |
| Match Instance Identifier:           gen26                             |
| Match Identifier:                    gen27                             |
+------------------------------------------------------------------------+
| Category               Discovery              Application              |
```

```
+------------------------------------------------------------------------+
| Name:                 DIAMOND-CVD-COATING                    GEARS      |
| Qualitative Value:             MEDIUM-LOW               MEDIUM-LOW      |
| Attribute Address:        DIAMOND-CVD-RHO                GEARS-RHO      |
+------------------------------------------------------------------------+


+------------------------------------------------------------------------+
|                              Match Table                               |
| Criterion Matched:                SPECIFIC-HEAT                         |
| Match Goodness:                          STRONG                         |
| Match Instance Identifier:                gen29                         |
| Match Identifier:                         gen30                         |
+------------------------------------------------------------------------+
| Category                 Discovery                Application           |
+------------------------------------------------------------------------+
| Name:                 DIAMOND-CVD-COATING                    GEARS      |
| Qualitative Value:                 MEDIUM                   MEDIUM      |
| Attribute Address:          DIAMOND-CVD-C                  GEARS-C      |
+------------------------------------------------------------------------+


+------------------------------------------------------------------------+
|                              Match Table                               |
| Criterion Matched:          THERMAL-CONDUCTIVITY                        |
| Match Goodness:                          STRONG                         |
| Match Instance Identifier:                gen31                         |
| Match Identifier:                         gen32                         |
+------------------------------------------------------------------------+
| Category                 Discovery                Application           |
+------------------------------------------------------------------------+
| Name:                 DIAMOND-CVD-COATING                    GEARS      |
| Qualitative Value:              VERY-HIGH                     HIGH      |
| Attribute Address:          DIAMOND-CVD-K                  GEARS-K      |
+------------------------------------------------------------------------+
```

C.1.2 Match Statistics

match_stats.out

```
+-----------------------------------------------------+
|                  Match Statistics                   |
|                                                     |
| Discovery:          DIAMOND-CVD-COATING             |
| Application:         HI-LOAD-ROLLER-BRG             |
+-----------------------------------------------------+
| Strong Matches:                                     |
|       LUB-COMPATIBILITY                             |
|              ROUGHNESS                              |
|                DENSITY                              |
|          SPECIFIC-HEAT                              |
```

```
+------------------------------------------------------+
| Weak Matches:                                        |
|     OXIDATION-RESISTANCE                             |
|                 HARDNESS                             |
|             POISSON-RATIO                            |
|             SHEAR-MODULUS                            |
|            YOUNGS-MODULUS                            |
|         USAGE-TEMPERATURE                            |
|         THERMAL-EXPANSION                            |
|      THERMAL-CONDUCTIVITY                            |
+------------------------------------------------------+
| Score = 0.33                                         |
+------------------------------------------------------+
```

```
+------------------------------------------------------+
|                Match Statistics                      |
|                                                      |
| Discovery:          DIAMOND-CVD-COATING              |
| Application:                      GEARS              |
+------------------------------------------------------+
| Strong Matches:                                      |
|                  DENSITY                             |
|            SPECIFIC-HEAT                             |
|      THERMAL-CONDUCTIVITY                            |
+------------------------------------------------------+
| Weak Matches:                                        |
+------------------------------------------------------+
| Score = 1.00                                         |
+------------------------------------------------------+
```

C.2 Output for Analysis Files

C.2.1 Application Files

gears.h

```
/* Begin APPLICATION Empirical Relationships */
    float GearXStart( int GearRatio, int NPinion, float phi );
    float GearTPinionStart( float X, int GearRatio,
       float phi, float psiP );
    float GearTPinion( float TPS, float f, float EEquiv,
       float omegaPinion, float CenterDistance, float KrhoC );
    float GearContactWidth( float TangentLoad, float EEquiv,
       float RadiusEquiv, float phi );
```

```
    float GearQPinion( float KrhoC, float DeltaTPinion,
       float aWidth, float VelocityPinion, float psiP );
    float GearCoatArea( float PitchDiameter, float FaceWidth );
    float GearEffectiveRadius( float CenterDistance, float X, float phi
);
    float GearEffectiveModulus( float EGear, float EPinion );
    float GearPinionPitchDia( float CenterDistance, int GearRatio );
    float GearPinionVelocity( float omegaPinion, float CenterDistance,
       float X, float phi );
    float GearGearVelocity( int GearRatio, float omegaPinion,
       float CenterDistance, float X, float phi );
    float GearSlideVelocity( float Vpinion, float Vgear );
    float GearRollVelocity( float Vpinion, float Vgear );
    float GearContactStress( float TangentLoad, float EEquiv,
       float RadiusEquiv, float phi );
    float GearBendingStress( float TangentLoad, int NPinion,
       float CenterDistance, int GearRatio );
    float GearTorque( float PitchDiameter, float TangentLoad );
/* End APPLICATION Empirical Relationships */
```

gearparams.h

```
/* Begin APPLICATION Design Parameters */
    float CenterDistance;
    int NPinion;
    int GearRatio;
    float omega;
    float TorqueIn;
/* End APPLICATION Design Parameters */
```

geardefines.h

```
/* Begin APPLICATION Constants */
    #DEFINE GEAR-TEMP-PINION-MAX-RISE *
    #DEFINE GEAR-TEMP-GEAR-MAX-RISE *
    #DEFINE GEAR-MAX-BENDING-STRESS *
    #DEFINE GEAR-MAX-CONTACT-STRESS *
/* End APPLICATION Constants */
```

C.2.2 Discovery Files

diamond_cvd.h

```
/* Begin DISCOVERY Empirical Relationships */
    float ThetaSSO( float PenDepth, float Length,
       float LayerConductivity, float SubstrateConductivity,
       float LayerThickness, float EntryLength );
    float ThetaOS( float PenDepth, float Length,
       float LayerConductivity, float SubstrateConductivity,
       float LayerThickness, float EntryLength );
    float ThetaS( float PenDepth, float Length,
       float LayerConductivity, float SubstrateConductivity );
    float ThetaSO( float ThetaSSO, float ThetaS );
    float ThetaO( float ThetaOS, float ThetaS );
    float Temperature( float Theta, float HeatIn );
    float PenetrationDepth( float Conductivity, float Density,
       float SpecificHeat, float Length, float Velocity );
    float EntryLength( float LayerConductivity, float LayerDensity,
       float LayerSpecificHeat, float Velocity, float LayerThickness );
    float NormalStress( float YoungsModulus, float ThermalExpansion,
       float TemperatureRise );
    float ShearStress( float LayerNormalStress, float
SubstrateNormalStress,
       float LayerThickness, float LayerWidth );
    float ThetaSS( float ThetaInterface, float ThetaSO_IFSS,
       float ThetaO_IFSS );
    float Delta( float ThetaUncoatedSubstrate, float ThetaSubstrate );
    float ThetaSS_P( float Delta, float ThetaSubstrate );
    float ThetaIF( float Delta, float Theta_IF );
    float ThetaD( float Delta, float Theta_D );
    float TemperatureS( float ThetaUncoatedSubstrate, float HeatIn );
    float TemperatureSubstrate( float ThetaSubstrate, float HeatIn );
    float TemperatureInterface( float ThetaInterfaceLayer, float HeatIn
);
    float TemperatureDiamond( float ThetaDiamond, float HeatIn );
/* End DISCOVERY Empirical Relationships */
```

diamond_cvd_defines.h

```
/* Begin DISCOVERY Constants */
/* End DISCOVERY Constants */
```

APPENDIX D
TECHNOLOGY ATTRIBUTE CLASS TEMPLATES

This appendix describes all the class templates developed for this study. The first section includes descriptions of the slots used in defining the various attribute classes. The next section is devoted to the class templates—starting with the highest level (root) classes which are used to develop the attribute hierarchy, and ending with the classes that are accessible to the user for creating instances.

D.1 Slot Descriptions

Table D-1. Slot descriptions for technology classes inheriting characteristics from TECHNOLOGY, LEVEL-1-ATTRIBUTE through LEVEL-4-ATTRIBUTE, and TECHNOLOGY-PROPERTY

Slot Name	Description
technology-name	unique identifier for the technology
technology-type	determines whether the instance is a DISCOVERY or APPLICATION
phylum	identifies a broad category for classifying a technology, e.g. a thermal barrier coating would have "SURFACE-TREATMENT" as its phylum
classification	the first-level specialization of a technology's phylum, e.g. "MATERIAL" or "MANUFACTURING"
order	the first-level specialization of the technology classification, e.g. "PHYSICAL," "ECONOMIC," "PROCESS"
family	the first-level specialization of the technology order, e.g. "THERMAL," "MECHANICAL," "PROCESS-LIMITS"
species	the most specific specialization of the technology attributes, e.g. "
assessment-criteria	determines whether the technology attribute is a capability, optional feature, desirable feature or need of the technology, e.g. high hardness is a "NEED" of a rolling element bearing
qualitative-value	a qualitative rating of a technology's property, e.g. "HIGH," "LOW"
qualitative-value-range	a minimum and maximum qualitative range of values for a technology's property, e.g. "MEDIUM" to "HIGH"
quantitative-value	magnitude of a technology's property, e.g. 1054.5
quantitative-value-range	a minimum and maximum magnitude for a technology's property, e.g. 853 to 1013
units	standard units of measurement for the magnitude of the technology's property, e.g. Newton/meter*meter for shear modulus
symbol	standard symbol for technology's property, e.g. K for thermal conductivity

Table D-2. Slot descriptions for classes inheriting characteristics from GOVERNING-EQUATION

Slot Name	Description
relationship-name	unique name for an empirical relationship describing a fundamental association amongst a technology's design parameters
c-header	a valid c language function prototype
function-name	unique identifier for an objective function to be used in optimizing some aspect of a technology's usage
parameter-name	a unique identifier for a design parameter
c-declaration	a valid c language variable declaration
c-type	a valid c language variable type, e.g. "int" or "float"
constant-name	a unique identifier for a constant
c-define	a valid c language #define for a constant
criteria-category	the general category associated with a qualitative requirement for an APPLICATION type technology, e.g. surface-contact
criteria-name	specific qualitative need for an APPLICATION type technology, e.g. slide-to-roll-ratio, contact-stress

Table D-3. Slot descriptions for classes inheriting characteristics from miscellaneous parent classes

Slot Name	Description
method-list	a list of methods used in a process associated with producing a product using the technology, e.g. "rough cut," "finish cut," "polish," and "inspect"
atmosphere-list	a list of atmospheres used in a process associated with a technology, e.g. argon, helium, air
maximum	the maximum attainable thickness attainable with a technolgy process that adds material
minimum	the minimum attainable thickness attainable with a technolgy process that adds material
criteria-category	the general category associated with a qualitative need for an APPLICATION type technology, e.g. surface-contact
criteria-name	specific qualitative need for an APPLICATION type technology, e.g. slide-to-roll-ratio, contact-stress
lub-list	a list of lubricants compatible with a technology's surface

D.2 Class Templates

D.2.1 High-Level Classes

The classes described here are not directly accessible to the user for creating instances. These classes build the underlying structure for the classes that are accessible to the user.

D.2.1.1 Root technology class

Table D-4. TECHNOLOGY class template

Class Name: TECHNOLOGY Parent Classes: USER Description: top-level class in the technology attribute hierarchy			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		

D.2.1.2 Abstract specialization classes

Table D-5. LEVEL-1-ATTRIBUTE class template

Class Name: LEVEL-1-ATTRIBUTE Parent Classes: USER Description: class to add a first-level attribute specialization to direct child classes of the TECHNOLOGY parent class			
Slot Name	Req'd?	Default Value	Allowed Values
classification			

Table D-6. LEVEL-2-ATTRIBUTE class template

Class Name: LEVEL-2-ATTRIBUTE Parent Classes: USER Description: class to add a first-level specialization to direct child classes of parents inheriting from the LEVEL-1-ATTRIBUTE class			
Slot Name	Req'd?	Default Value	Allowed Values
order			

Table D-7. LEVEL-3-ATTRIBUTE class template

Class Name: LEVEL-3-ATTRIBUTE
Parent Classes: USER
Description: class to add a first-level specialization to direct child classes of parents inheriting from the LEVEL-2-ATTRIBUTE class

Slot Name	Req'd?	Default Value	Allowed Values
family			

Table D-8. LEVEL-4-ATTRIBUTE class template

Class Name: LEVEL-4-ATTRIBUTE
Parent Classes: USER
Description: class to add a first-level specialization to direct child classes of parents inheriting from the LEVEL-3-ATTRIBUTE class; determines whether attribute is requirement (NEED) or capability of a technology

Slot Name	Req'd?	Default Value	Allowed Values
species			
assessment-criteria		CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY

Table D-9. TECHNOLOGY-PROPERTY class template

Class Name: TECHNOLOGY-PROPERTY
Parent Classes: USER
Description: class to describe the qualitative and quantitative characteristics, standard units and symbol of a technology's property

Slot Name	Req'd?	Default Value	Allowed Values
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units			
symbol			

Table D-10. MFG-PROCESS-METHOD-LIST class template

Class Name: MFG-PROCESS-METHOD-LIST Parent Classes: USER Description: class to add a multiple entry list structure to store manufacturing methods			
Slot Name	Req'd?	Default Value	Allowed Values
species			
method-list			

D.2.1.3 First-level specializations of the root technology class

Table D-11. MATERIAL-ATTRIBUTE class template

Class Name: MATERIAL-ATTRIBUTE Parent Classes: TECHNOLOGY, LEVEL-1-ATTRIBUTE Description: first-level TECHNOLOGY specialization class containing material attribute subclasses			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	

Table D-12. MANUFACTURING-ATTRIBUTE class template

Class Name: MANUFACTURING-ATTRIBUTE Parent Classes: TECHNOLOGY, LEVEL-1-ATTRIBUTE Description: first-level TECHNOLOGY specialization class containing manufacturing attribute subclasses			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	

Table D-13. OTHER-ATTRIBUTE class template

Class Name: OTHER-ATTRIBUTE
Parent Classes: TECHNOLOGY, LEVEL-1-ATTRIBUTE
Description: first-level TECHNOLOGY specialization class containing other attribute subclasses

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		OTHER	

Table D-14. GOVERNING-EQUATION class template

Class Name: GOVERNING-EQUATION
Parent Classes: TECHNOLOGY, LEVEL-1-ATTRIBUTE
Description: first-level TECHNOLOGY specialization class containing governing equation subclasses

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		EQUATION	

D.2.1.4 Second-level specializations of the root technology class

Table D-15. PHYSICAL-PROPERTY class template

Class Name: PHYSICAL-PROPERTY
Parent Classes: MATERIAL-ATTRIBUTE, LEVEL-2-ATTRIBUTE
Description: first-level MATERIAL-ATTRIBUTE specialization class containing physical property attribute subclasses

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	

Table D-16. ECONOMIC-PROPERTY class template

Class Name: ECONOMIC-PROPERTY Parent Classes: MATERIAL-ATTRIBUTE, LEVEL-2-ATTRIBUTE Description: first-level MATERIAL-ATTRIBUTE specialization class containing economic property attribute subclasses			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		ECONOMIC	

Table D-17. OTHER-PROPERTY class template

Class Name: OTHER-PROPERTY Parent Classes: MATERIAL-ATTRIBUTE, LEVEL-2-ATTRIBUTE Description: first-level MATERIAL-ATTRIBUTE specialization class containing other property attribute subclasses			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		OTHER	

Table D-18. PROCESS-PROPERTY class template

Class Name: PROCESS-PROPERTY Parent Classes: MANUFACTURING-ATTRIBUTE, LEVEL-2-ATTRIBUTE Description: first-level MANUFACTURING-ATTRIBUTE specialization class containing process property attribute subclasses			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC- TURING	
order		PROCESS	

D.2.1.5 Third-level specializations of the root technology class

Table D-19. THERMAL-PROPERTY class template

Class Name: THERMAL-PROPERTY Parent Classes: PHYSICAL-PROPERTY, LEVEL-3-ATTRIBUTE Description: first-level PHYSICAL-PROPERTY specialization class defining thermal property sub-classes for a technology			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		THERMAL	

Table D-20. MECHANICAL-PROPERTY class template

Class Name: MECHANICAL-PROPERTY Parent Classes: PHYSICAL-PROPERTY, LEVEL-3-ATTRIBUTE Description: first-level PHYSICAL-PROPERTY specialization class defining mechanical property sub-classes for a technology			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		MECHAN- ICAL	

Table D-21. ENVIRONMENTAL-PROPERTY class template

Class Name: ENVIRONMENTAL-PROPERTY Parent Classes: PHYSICAL-PROPERTY, LEVEL-3-ATTRIBUTE Description: first-level PHYSICAL-PROPERTY specialization class defining environmental property sub-classes for a technology			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		ENVIRON-MENTAL	

Table D-22. PROCESS-ECONOMICS class template

Class Name: PROCESS-ECONOMICS Parent Classes: PROCESS-PROPERTY, LEVEL-3-ATTRIBUTE Description: first-level PROCESS-PROPERTY specialization class defining process economics property sub-classes for a technology			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		ECONOMIC	

Table D-23. PROCESS-PARAMETER class template

Class Name:	PROCESS-PARAMETER
Parent Classes:	PROCESS-PROPERTY, LEVEL-3-ATTRIBUTE
Description:	first-level PROCESS-PROPERTY specialization class defining process parameter sub-classes for a technology

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		PARAMETER	

Table D-24. PROCESS-METHOD class template

Class Name:	PROCESS-METHOD
Parent Classes:	PROCESS-PROPERTY, LEVEL-3-ATTRIBUTE
Description:	first-level PROCESS-PROPERTY specialization class defining process method sub-classes for a technology

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		METHOD	

Table D-25. PROCESS-LIMITS class template

Class Name: PROCESS-LIMITS Parent Classes: PROCESS-PROPERTY, LEVEL-3-ATTRIBUTE Description: first-level PROCESS-PROPERTY specialization class defining process limits sub-classes for a technology			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFACTURING	
order		PROCESS	
family		LIMITS	

D.2.2 Classes Accessible to the User for Instances

The classes described in this section are available to the user to create instances for manipulation in the knowledge base.

D.2.2.1 Specializations of the GOVERNING-EQUATION class

Table D-26. EMPIRICAL-RELATIONSHIP class template

Class Name: EMPIRICAL-RELATIONSHIP Parent Classes: GOVERNING-EQUATION Description: first-level GOVERNING-EQUATION specialization class defining empirical relationships that model a technology's behavior in the c programming language—intended to be used in conjunction with a library of pre-defined c implementations of pertinent functions			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		EQUATION	
relationship-name	yes		
c-header	yes		

Table D-27. OBJECTIVE-FUNCTION class template

Class Name: OBJECTIVE-FUNCTION
Parent Classes: GOVERNING-EQUATION
Description: first-level GOVERNING-EQUATION specialization class defining optimzation objective functions for designing applications of a technology in the c programming language—intended to be used in conjunction with a library of pre-defined c implementations of pertinent functions

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		EQUATION	
function-name	yes		
c-header	yes		

Table D-28. DESIGN-PARAMETER class template

Class Name: DESIGN-PARAMETER
Parent Classes: GOVERNING-EQUATION
Description: first-level GOVERNING-EQUATION specialization class defining design parameter variables for a technology in the c programming language

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		EQUATION	
parameter-name	yes		
c-declaration	yes		
c-type	yes		"int"|"short"|"long"|"float"| "double"|"long double"

Table D-29. CONSTANT class template

Class Name: CONSTANT Parent Classes: GOVERNING-EQUATION Description: first-level GOVERNING-EQUATION specialization class defining design or analysis constants used for a technology in the c programming language			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		EQUATION	
constant-name	yes		
c-define	yes		
c-type	yes		"int"|"short"|"long"|"float"| "double"|"long double"

D.2.2.2 The QUALITATIVE-REQTS class

Table D-30. QUALITATIVE-REQTS class template

Class Name: QUALITATIVE-REQTS Parent Classes: TECHNOLOGY, LEVEL-1-ATTRIBUTE Description: first-level TECHNOLOGY specialization class containing qualitative requirements of APPLICATION technology types			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		APPLICATION	DISCOVERY|APPLICATION
phylum	yes		
classification		QUALITATIVE-REQTS	
criteria-category	yes		
criteria-name	yes		
qualitative-value	yes		VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH

D.2.2.3 Specializations of the THERMAL-PROPERTY class

Table D-31. THERMAL-CONDUCTIVITY class template

Class Name:	THERMAL-CONDUCTIVITY
Parent Classes:	TECHNOLOGY-PROPERTY, THERMAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the THERMAL-PROPERTY class defining the thermal conductivity property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		THERMAL	
species		HEAT-TRANSFER	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		Watt/meter-deg-K	
symbol		K	

Table D-32. SPECIFIC-HEAT class template

Class Name: SPECIFIC-HEAT Parent Classes: TECHNOLOGY-PROPERTY, THERMAL-PROPERTY, LEVEL-4-ATTRIBUTE Description: first-level specialization of the THERMAL-PROPERTY class defining the specific heat property; direct instances of this class can be created			
Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		THERMAL	
species		HEAT-TRANSFER	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		Joule/kilogram-deg-K	
symbol		C	

Table D-33. THERMAL-EXPANSION class template

Class Name:	THERMAL-EXPANSION
Parent Classes:	TECHNOLOGY-PROPERTY, THERMAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the THERMAL-PROPERTY class defining the thermal expansion coefficient property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		THERMAL	
species		HEAT-TRANSFER	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		meter/meter-deg-K	
symbol		ALPHA	

Table D-34. THERMAL-DIFFUSIVITY class template

Class Name:	THERMAL-DIFFUSIVITY
Parent Classes:	TECHNOLOGY-PROPERTY, THERMAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the THERMAL-PROPERTY class defining the thermal diffusivity property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		THERMAL	
species		HEAT-TRANSFER	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		meter*meter/sec	
symbol		ALPHA-sub-diff	

Table D-35. THERMAL-SHOCK class template

Class Name:	THERMAL-SHOCK
Parent Classes:	TECHNOLOGY-PROPERTY, THERMAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the THERMAL-PROPERTY class defining the thermal shock property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		THERMAL	
species		HEAT-TRANSFER	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		Watt/meter-sec	
symbol		Th-sub-shock	

Table D-36. USAGE-TEMPERATURE class template

Class Name:	USAGE-TEMPERATURE
Parent Classes:	TECHNOLOGY-PROPERTY, THERMAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the THERMAL-PROPERTY class defining the maximum usage temperature property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		THERMAL	
species		HEAT-TRANSFER	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		deg-K	
symbol		T-sub-max	

Table D-37. MELT-POINT class template

Class Name:	MELT-POINT
Parent Classes:	TECHNOLOGY-PROPERTY, THERMAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the THERMAL-PROPERTY class defining the melting point property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		THERMAL	
species		HEAT-TRANSFER	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		deg-K	
symbol		T-sub-melt	

D.2.2.4 Specializations of the MECHANICAL-PROPERTY class

Table D-38. YOUNGS-MODULUS class template

Class Name: YOUNGS-MODULUS
Parent Classes: TECHNOLOGY-PROPERTY, MECHANICAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description: first-level specialization of the MECHANICAL-PROPERTY class defining the Young's modulus property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		MECHAN-ICAL	
species		STRESS-STRAIN	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		Newton/meter*meter	
symbol		E	

Table D-39. SHEAR-MODULUS class template

Class Name:	SHEAR-MODULUS
Parent Classes:	TECHNOLOGY-PROPERTY, MECHANICAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the MECHANICAL-PROPERTY class defining the shear modulus property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		MECHAN-ICAL	
species		STRESS-STRAIN	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		Newton/ meter*meter	
symbol		G	

Table D-40. POISSON-RATIO class template

Class Name:	POISSON-RATIO
Parent Classes:	TECHNOLOGY-PROPERTY, MECHANICAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the MECHANICAL-PROPERTY class defining the Poisson ratio property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		MECHAN-ICAL	
species		STRESS-STRAIN	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units			
symbol		NU	

Table D-41. DENSITY class template

Class Name:	DENSITY
Parent Classes:	TECHNOLOGY-PROPERTY, MECHANICAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the MECHANICAL-PROPERTY class defining the density property (standard conditions); direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		MECHAN-ICAL	
species		MASS-VOLUME	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		kilogram/ meter*meter* meter	
symbol		RHO	

Table D-42. FRICTION-COEFFICIENT-DRY class template

Class Name:	FRICTION-COEFFICIENT-DRY
Parent Classes:	TECHNOLOGY-PROPERTY, MECHANICAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the MECHANICAL-PROPERTY class defining the dry coefficient of friction property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		MECHAN-ICAL	
species		FRICTION	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units			
symbol		MU-sub-dry	

Table D-43. FRICTION-COEFFICIENT-LUB class template

Class Name: FRICTION-COEFFICIENT-LUB Parent Classes: TECHNOLOGY-PROPERTY, MECHANICAL-PROPERTY, LEVEL-4-ATTRIBUTE Description: first-level specialization of the MECHANICAL-PROPERTY class defining the lubricated coefficient of friction property; direct instances of this class can be created			
Slot Name	**Req'd?**	**Default Value**	**Allowed Values**
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		MECHAN-ICAL	
species		FRICTION	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units			
symbol		MU-sub-lub	

Table D-44. ROUGHNESS class template

Class Name:	ROUGHNESS
Parent Classes:	TECHNOLOGY-PROPERTY, MECHANICAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the MECHANICAL-PROPERTY class defining the surface roughness property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		MECHAN-ICAL	
species		FRICTION	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		meter	
symbol		R	

Table D-45. HARDNESS class template

Class Name:	HARDNESS
Parent Classes:	TECHNOLOGY-PROPERTY, MECHANICAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the MECHANICAL-PROPERTY class defining the Knoop hardness property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		MECHAN-ICAL	
species		STRESS-STRAIN	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		meter	
symbol		R	

Table D-46. TOUGHNESS class template

Class Name:	TOUGHNESS
Parent Classes:	TECHNOLOGY-PROPERTY, MECHANICAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the MECHANICAL-PROPERTY class defining the toughness property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		MECHAN-ICAL	
species		STRESS-STRAIN	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		Newton/ meter*meter	
symbol		TNS	

Table D-47. CRACK-RESISTANCE class template

Class Name:	CRACK-RESISTANCE
Parent Classes:	TECHNOLOGY-PROPERTY, MECHANICAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the MECHANICAL-PROPERTY class defining the crack resistance property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		MECHAN-ICAL	
species		FRACTURE	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		Newton/ meter*meter	
symbol		CR	

D.2.2.5 Specializations of the ENVIRONMENTAL-PROPERTY class

Table D-48. LUB-FILM-THKNS class template

Class Name: LUB-FILM-THKNS
Parent Classes: TECHNOLOGY-PROPERTY, ENVIRONMENTAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description: first-level specialization of the ENVIRONMENTAL-PROPERTY class defining the lubricating film thickness property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		ENVIRON-MENTAL	
species		HYDRO-DYNAMICS	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		meter	
symbol		h-sub-lub	

Table D-49. LUB-COMPATIBILITY class template

Class Name:	LUB-COMPATIBILITY
Parent Classes:	TECHNOLOGY-PROPERTY, ENVIRONMENTAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the ENVIRONMENTAL-PROPERTY class defining the compatible lubricating film list property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		ENVIRON-MENTAL	
species		HYDRO-DYNAMICS	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units			
symbol			
lub-list			

Table D-50. OXIDATION-RESISTANCE class template

Class Name: OXIDATION-RESISTANCE
Parent Classes: TECHNOLOGY-PROPERTY, ENVIRONMENTAL-PROPERTY, LEVEL-4-ATTRIBUTE
Description: first-level specialization of the ENVIRONMENTAL-PROPERTY class defining the oxidation resistance property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MATERIAL	
order		PHYSICAL	
family		ENVIRON-MENTAL	
species		OXIDATION	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units			
symbol			

D.2.2.6 Specialization of the PROCESS-ECONOMICS class

Table D-51. MATERIAL-COST class template

Class Name:	MATERIAL-COST		
Parent Classes:	TECHNOLOGY-PROPERTY, PROCESS-ECONOMICS, LEVEL-4-ATTRIBUTE		
Description:	first-level specialization of the PROCESS-ECONOMICS class defining the raw material cost property; direct instances of this class can be created		
Slot Name	**Req'd?**	**Default Value**	**Allowed Values**
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		ECONOMIC	
species		COST-PER-GRAM	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		dollars/gram	
symbol		C-sub-M	

Table D-52. CAPITAL-EQUIPMENT-COST class template

Class Name:	CAPITAL-EQUIPMENT-COST
Parent Classes:	TECHNOLOGY-PROPERTY, PROCESS-ECONOMICS, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the PROCESS-ECONOMICS class defining the capital equipment cost property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		ECONOMIC	
species		COST-PER-GRAM	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		dollars/gram	
symbol		C-sub-CE	

Table D-53. LABOR-COST class template

Class Name:	LABOR-COST
Parent Classes:	TECHNOLOGY-PROPERTY, PROCESS-ECONOMICS, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the PROCESS-ECONOMICS class defining the labor cost property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		ECONOMIC	
species		COST-PER-GRAM	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		dollars/gram	
symbol		C-sub-L	

Table D-54. POWER-COST class template

Class Name:	POWER-COST
Parent Classes:	TECHNOLOGY-PROPERTY, PROCESS-ECONOMICS, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the PROCESS-ECONOMICS class defining the power cost property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFACTURING	
order		PROCESS	
family		ECONOMIC	
species		COST-PER-GRAM	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		dollars/gram	
symbol		C-sub-P	

Table D-55. OVERHEAD-COST class template

Class Name: OVERHEAD-COST
Parent Classes: TECHNOLOGY-PROPERTY, PROCESS-ECONOMICS, LEVEL-4-ATTRIBUTE
Description: first-level specialization of the PROCESS-ECONOMICS class defining the overhead cost property; direct instances of this class can be created

<table>
<tr><th>Slot Name</th><th>Req'd?</th><th>Default Value</th><th>Allowed Values</th></tr>
<tr><td>technology-name</td><td>yes</td><td></td><td></td></tr>
<tr><td>technology-type</td><td></td><td>DISCOVERY</td><td>DISCOVERY|APPLICATION</td></tr>
<tr><td>phylum</td><td>yes</td><td></td><td></td></tr>
<tr><td>classification</td><td></td><td>MANUFAC-TURING</td><td></td></tr>
<tr><td>order</td><td></td><td>PROCESS</td><td></td></tr>
<tr><td>family</td><td></td><td>ECONOMIC</td><td></td></tr>
<tr><td>species</td><td></td><td>COST-PER-GRAM</td><td></td></tr>
<tr><td>assessment-criteria</td><td>yes</td><td>CAPABILITY</td><td>NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY</td></tr>
<tr><td>qualitative-value</td><td></td><td></td><td>VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH</td></tr>
<tr><td>qualitative-value-range</td><td></td><td></td><td></td></tr>
<tr><td>quantitative-value</td><td></td><td></td><td></td></tr>
<tr><td>quantitative-value-range</td><td></td><td></td><td></td></tr>
<tr><td>units</td><td></td><td>dollars/gram</td><td></td></tr>
<tr><td>symbol</td><td></td><td>C-sub-O</td><td></td></tr>
</table>

D.2.2.7 Specialization of the PROCESS-PARAMETER class

Table D-56. PROCESS-PRESSURE class template

Class Name: PROCESS-PRESSURE
Parent Classes: TECHNOLOGY-PROPERTY, PROCESS-PARAMETER, LEVEL-4-ATTRIBUTE
Description: first-level specialization of the PROCESS-PARAMETER class defining the process pressure property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		PARAMETER	
species		PRESSURE	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		Newton/ meter*meter	
symbol		P-sub-p	

Table D-57. PROCESS-ATMOSPHERE class template

Class Name:	PROCESS-ATMOSPHERE
Parent Classes:	TECHNOLOGY-PROPERTY, PROCESS-PARAMETER, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the PROCESS-PARAMETER class defining the process atmosphere list property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		PARAMETER	
species		ATMO-SPHERE	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units			
symbol			
atmosphere-list			

Table D-58. PROCESS-SURFACE-TEMPERATURE class template

Class Name:	PROCESS-SURFACE-TEMPERATURE
Parent Classes:	TECHNOLOGY-PROPERTY, PROCESS-PARAMETER, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the PROCESS-PARAMETER class defining the process surface temperature property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		PARAMETER	
species		TEMPER-ATURE	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		deg-C	
symbol		T-sub-surf	

D.2.2.8 Specialization of the PROCESS-LIMITS class

Table D-59. PROCESS-LIMITS-THKNS class template

Class Name:	PROCESS-LIMITS-THKNS
Parent Classes:	TECHNOLOGY-PROPERTY, PROCESS-LIMITS, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the PROCESS-LIMITS class defining the process maximum and minimum thickness property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		LIMITS	
species		LENGTH	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		meter	
symbol		h-sub-0	
maximum			
minimum			

Table D-60. PROCESS-LIMITS-SPATIAL class template

Class Name:	PROCESS-LIMITS-SPATIAL
Parent Classes:	TECHNOLOGY-PROPERTY, PROCESS-LIMITS, LEVEL-4-ATTRIBUTE
Description:	first-level specialization of the PROCESS-LIMITS class defining the process volume requirements for equipment property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		LIMITS	
species		VOLUME	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEA-TURE|OPTIONAL-FEA-TURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		meter*meter*meter	
symbol		V-sub-sp	

Table D-61. PROCESS-LIMITS-DEPOSITION class template

Class Name: PROCESS-LIMITS-DEPOSITION
Parent Classes: TECHNOLOGY-PROPERTY, PROCESS-LIMITS, LEVEL-4-ATTRIBUTE
Description: first-level specialization of the PROCESS-LIMITS class defining the process deposition rate property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		LIMITS	
species		DEPOSITION-RATE	
assessment-criteria	yes	CAPABILITY	NEED|DESIRABLE-FEATURE|OPTIONAL-FEATURE|CAPABILITY
qualitative-value			VERY-LOW|LOW|MEDIUM-LOW|MEDIUM|MEDIUM-HIGH|HIGH|VERY-HIGH
qualitative-value-range			
quantitative-value			
quantitative-value-range			
units		gram/hour	
symbol		M-sub-t	

D.2.2.9 Specialization of thePROCESS-METHOD class

Table D-62. METHOD-PREPARATION class template

Class Name:	METHOD-PREPARATION
Parent Classes:	MFG-PROCESS-METHOD-LIST, PROCESS-METHOD
Description:	first-level specialization of the PROCESS-METHOD class defining the process preparartion methods list property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		METHOD	
species		PREPARA-TION	
method-list			

Table D-63. METHOD-APPLICATION class template

Class Name:	METHOD-APPLICATION
Parent Classes:	MFG-PROCESS-METHOD-LIST, PROCESS-METHOD
Description:	first-level specialization of the PROCESS-METHOD class defining the process application methods list property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		METHOD	
species		APPLICA-TION	
method-list			

Table D-64. METHOD-FINISHING class template

Class Name:	METHOD-FINISHING
Parent Classes:	MFG-PROCESS-METHOD-LIST, PROCESS-METHOD
Description:	first-level specialization of the PROCESS-METHOD class defining the process finishing methods list property; direct instances of this class can be created

Slot Name	Req'd?	Default Value	Allowed Values
technology-name	yes		
technology-type		DISCOVERY	DISCOVERY|APPLICATION
phylum	yes		
classification		MANUFAC-TURING	
order		PROCESS	
family		METHOD	
species		FINISHING	
method-list			

REFERENCES

Angu94 Angus, J. "Development of Low-Pressure Diamond Growth in the United States." In: *Synthetic Diamond: Emerging CVD Science and Technology*, edited by Spear, K. & Dismukes, J., New York: John Wiley & Sons, Inc., 1994, p. 21-39.

ASLE90 American Society for Lubrication Engineers, *General Lubrication Engineering Practice*, New York: ASLE, 1990.

ASME81 American Society of Mechanical Engineers Research Committee on Lubrication, *Strategy for Energy Conservation through Tribology*, New York: ASME, 1981.

Barb85 Barbuceanu, M. "An Object-Centered Framework for Expert Systems in Computer-Aided Design." In: *Knowledge Engineering in Computer-Aided Design*, edited by Gero, J.S., New York: Elsevier Science Publishing Company, Inc., 1985, p. 223-258.

Baro90 Baron, S. "Overcoming Barriers to Technology Transfer." *Research • Technology Management* 33.1:38-43, 1990.

Bell57 Bellman, R. *Dynamic Programming*, Princeton: Princeton University Press, 1957.

Blok37 Blok, H., "Theoretical Study of Temperature Rise at Sufaces of Actual Contact under Oiliness Lubricating Conditions," *Proceedings of the General Discussion on Lubrication*, London: Institute of Mechanical Engineers, Part 2, 1937, p. 222.

Booc91 Booch, G. *Object Oriented Design With Applications*, Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc., 1991.

Brow85 Brown, D.C. & Chandrasekaran, B. "Expert Systems for a Class of Mechanical Design Activity." In: *Knowledge Engineering in Computer-Aided Design*, edited by Gero, J.S., New York: Elsevier Science Publishing Company, Inc., 1985, p. 259-290.

Busc87 Busch, J. *Technical Cost Modeling of Plastics Fabrication Processes*, Thesis, Cambridge: Massachusetts Institute of Technology, 1987.

Busc94 Busch, J. & Dismukes, J. "A Comparative Assessment of CVD Diamond Manufacturing Technology and Economics." In: *Synthetic Diamond: Emerging CVD Science and Technology*, edited by Spear, K. & Dismukes, J., New York: John Wiley & Sons, Inc., 1994, p. 581-624.

Chal76 Challa, K. & Berra, P. "Automated Planning and Machining Procedures—A Systems Approach." *Computers in Industrial Engineering* 1:35-46, 1976.

CIME87 "A New Age for Accounting." *Computers in Mechanical Engineering* July/August:12-15, 1987.

Coop88 Cooper, R. "The Rise of Activity-Based Costing—Part One: What Is an Activity-Based Cost System?" *Journal of Cost Management* Summer:45-53, 1988.

Cors85 Corser, T. & Seireg, A. "Optimizing a Design for Production, Inspection and Operation." *Computers in Mechanical Engineering* September:18-27, 1985.

Davi93 *Diamond Films and Coatings: Development, Properties, and Applications*, edited by Davis, R., Park Ridge, NJ: Noyes Publications, 1993.

Doon95 Dooner, D. & Seireg, A. *The Kinematic Geometry of Gearing: a Concurrent Engineering Approach*, New York: John Wiley & Sons, Inc., 1995. 450 pp.

Fiel92 *The Properties of Natural and Synthetic Diamond*, edited by Field, J. London: Academic Press/Harcourt Brace Jovanovich, 1992.

Fish87 Fisher, W.H. "Tuning Input/Output Analysis to Technology." *Mechanical Engineering* 109(1):46-49, 1987.

Forg82 Forgy, C. "Rete: A Fast Algorithm for the Many Pattern/Many Objects Match Problem." *Artificial Intelligence* 19:1:17-37, 1982.

Full84 Fuller, D. *Theory and Practice of Lubrication for Engineers*, New York: John Wiley & Sons, 1984.

Gadh93 Gadh, R. & Novak, J. "Cin-Con: An Intelligent Feature-based Concurrent Product and Process Design System." *Proceedings of JSME-ASME Joint Workshop on Engineering Design '93: "Frontiers in Engineering Design,"* Tokyo, 88-93, 1993.

Giar91 Giarratano, J. *CLIPS User's Guide*, Houston: NASA Johnson Space Center I.S.D. Software Technology Branch, 1991. Ed. 5

Giar93 Giarrantano, J. *CLIPS Basic Programming Guide Version 6.0*, Houston: NASA Johnson Space Center I.S.D. Software Technology Branch, 1993, 350 pp.

Gibs91 Gibson, D. & Niwa, K. "Knowledge-Based Technology Transfer." In: *Technology Management: The New International Language*, edited by Kocaoglu, D. & Niwa, K., New York: IEEE, 1991, p. 503-506.

Gonz93 Gonzalez, A. & Dankel, D.,II *Engineering Knowledge-based Systems: Theory and Practice*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

Hnat93 Hnatio, J. *Program for Technology Stabilization and Enterprise Development in the New Independent States of the Former Soviet Union*, 1993. (UnPub)

Jaeg42 Jaeger, J., "Moving sources of Heat and the Temperature at Sliding Contacts," *Proceedings of the Royal Society*, London: N.S.W., 56:203, 1942.

Jost91 Jost, H. "TRIBOLOGY: The First 25 Years and Tasks for the Future." *Proceedings of the 55th Anniversary Conference of the Chinese Mechanical Engineering Society: "World Progress on Mechanical Engineering,"* Beijing, 128-162, 1991.

Kino93 Kinoglu, F., Riley, D. & Donath, M. "Expert System Model of the Design Process". In: *Smart Manufacturing With Artificial Intelligence*, edited by Krakauer, J. Dearborn: Computer and Automated Systems Association of SME, 1987, p. 37-48.

Leon87 Leontief, W. "The Ins and Outs of Input/Output Analysis." *Mechanical Engineering* 109 (1):28-35, 1987.

Li89 Li, Y. & Seireg, A. "Predicting the Coefficient of Friction in Sliding-Rolling Contacts." *ASME Journal of Tribology* 111:386-390, 1989.

Lin85 Lin, T. & Seireg, A. "An Optimum Algorithm for Gear Systems Incorporating Surface Temperature." ASME Journal of Mechanisms, Transmissions, & Automation in Design, 107:549-555, 1985.

Liu87 Liu, S. "An Intelligent Framework for Engineering Decision Making." *Proceedings of SAE International Congress and Exposition* 1987.

Luka88a Lukas, M.P. & Meerbaum, M.I. "Improving Productivity by Capturing and Automating Design/Manufacturing Knowledge." *The Journal of Manufacturing* 1(1):1988.

Lux93 Lux, B. & Haubner, R. "Diamond as a Wear-Resistant Coating." In: *Diamond Films and Coatings: Development, Properties, and Applications*, edited by Davis, R. Park Ridge, NJ: Noyes Publishers, 1993, p. 183-243.

Luxh94 Luxhoj, J. "Evaluation of Alternative Transition Strategies for Advanced Technology Systems." *The International Journal of Human Factors in Manufacturing* 4:81-94, 1994.

Mans65 Manson, S.S. "Fatigue, A Complex Subject—Some Simple Approximations." *Experimental Mechanics* 5(7):193-226, 1965.

Mous94 Moustakas, T. "Growth of Diamond by CVD Methods and Effects of Process Parameters." In: *Synthetic Diamond: Emerging CVD Science and Technology*, edited by Spear, K. & Dismukes, J., New York: John Wiley & Sons, Inc., 1994, p. 145-192.

O'Do66 O'Donoghue, J. & Cameron, A. "Friction and Temperature in Rolling Sliding Contacts." *ASLE Transactions* 9:186-194, 1966.

Orth90 Orthwein, W. "Clutches and Brakes." In: *Machine Component Design*, edited by Dunn, V., St. Paul: West Publishing Company, 1990, p. 645-708.

Othm89 Othman, M. & Seireg, A. "A Procedure for Evaluating the Friction Properties of Hertzian Contacts Under Reciprocating Sliding Motion." *ASME Journal of Tribology* 1-4, 1989.

Pati77 Patir, N. & Cheng, H.S. "Prediction of Bulk Temperature in Spur Gears Based on Finite Element Temperature Analysis" *ASLE* Preprint No. 77-LC-3B-2, Oct. 1977.

Plan93 Plano, L. & Pinneo, M. "Non-Electric Applications of Diamond Films." In: *Diamond Films and Coatings: Development, Properties, and Applications*, edited by Davis, R. Park Ridge, NJ: Noyes Publishers, 1993, p. 339-380.

PNWL85 Pacific Northwest Laboratory "Tribological Sinks—Money Down the Drain." Pacific Northwest Laboratory Report PNL-SA-12865, Washington: DOE, 1985.

Ra91 Ra, J. "Hierarchical Decision Process." In: *Technology Management: The New International Language*, edited by Kocaoglu, D. & Niwa, K., New York: IEEE, 1991, p. 595-599.

Raja89 Rajab, M.D. & Seireg, A.A. "Predicting Life of Composites from Static Tests." In: *Proceedings of the First USSR—US Symposium on Mechanics of Composite Materials*, edited by Tsai, S.W., Whitney, J.M., Chou, T.W. & Jones, R.M., New York: ASME, 1989, p. 49-56.

Raom58 Raomondi, A. & Boyd, J. "A Solution for Finite Journal Bearing and Its Application to Analysis and Design, Parts I, II and III." *Transactions of American Society of Lubrication Engineers* 1(1):159-209, 1958.

Rash86 Rashid, M. & Seireg, A. "Heat Partition and Transient Temperature Distribution in Layered Concentrated Contacts. Part II—Dimensionless Relationships and Numerical Results." *ASME Journal of Tribology* 102-107, 1986.

Rej94 Rej, D. & Alexander, R. "Cost Estimate for Commercial Plasma Source Ion Implantation." J. Vac. Sci. Technol. B, 12(4):2380-2387, 1994.

Remi91 Reminger, B. "How to Manage Technology by an Expert System." In: *Technology Management: The New International Language*, edited by Kocaoglu, D. & Niwa, K., New York: IEEE, 1991, p. 554-561.

Rich91 Rich, E. & Knight, K. *Artificial Intelligence*, New York: McGraw-Hill, 1991. Ed. 2

Saat90 Saaty, T. *Decision Making for Leaders*, Pittsburgh: RWS Publications, 1990. Ed. 2, 292 pp.

Sage89 Sage, A. "Systems Management of Emerging Technologies." *Information and Decision Technologies* 15:307-325, 1989.

Sahn89 Sahni, S. *Software Development in Pascal*, North Oaks, MN: The Camelot Publishing Co., 1989. Ed. 2, 690 pp.

Seir69 Seireg, A. & Ezzat, H. "Optimum Design of Hydrodynamic Journal Bearings." *ASME Journal of Lubrication Technology* 91(July):516-523, 1969.

Seir70 Seireg, A. *Mechanical Systems Design: Creative Synthesis and Optimization*, Madison: University of Wisconsin, 1970.

Seir73 Seireg, A. & Ezzat, H. "Thermo-hydrodynamic Phenomena in Fluid Film Bearing." *ASME Journal of Lubrication Technology* 95:187-194, 1973.

Seir81 Seireg, A. & Hsue, E. "An Experimental Investigation of the Effect of Lubricant Properties on Temperature and Wear in Sliding Concentrated Contacts." *ASME Journal of Lubrication Technology* 103(2):261-265, 1981.

Seir82 Seireg, A. & Dandage, S. "Empiracle Design Procedures for Thermo-hydrodynamic Behavior of Journal Bearings." *ASME Journal of Lubrication Technology* 104:135-148, 1982.

Seir87 Seireg, A.A. "Establishing the Market Value of Innovation." *Mechanical Engineering* 109(1):36-39, 1987.

Seir95 Seireg, A. "Retrofit Engineering: A Methodology for Conversion." In: *Defense Conversion*, edited by Green, A., Hampton, VA: A. Deepak Publishing Co., 1995 (in press).

Shig83 Shigley, J. & Mitchell, L. "Clutches, Brakes, Couplings, and Flywheels." In: *Mechanical Engineering Design*, edited by Klas, R. & Eichberg, M., New York: McGraw-Hill, 1983, p. 714-754.

Somm92 Sommerville, I. *Software Engineering*, Wokingham, England: Addison-Wesley, 1992. Ed. 4

Spea94 *Synthetic Diamond: Emerging CVD Science and Technology*, edited by Spear, K. & Dismukes, New York: John Wiley & Sons, Inc., 1994. 663 pp.

Tand93 Tandon, M. *An Integrated Design Decision System for Optimum Life-cycle Cost with Emphasis on Constraint Management*, Doctoral Dissertation, Gainesville: University of Florida, 1993.

Tew87 Tew, P. *Knowledge-based Expert Systems Technology and Basic Database Programming Techniques for "CIME-ISE" System*, Madison: University of Wisconsin, 1987.

TRIB73 *Tribology Handbook*, London:Butterworths, 1973.

Turn90 Turney, P. "Using Activity-Based Costing to Achieve Manufacturing Excellence." In: *Emerging Practices in Cost Management*, Boston: Warren, Gorham, Lamont, 1990, p. D4-1-D4-9.

Vand84 Vanderplaats, G. *Numerical Optimization Techniques for Engineering Design: With Applications*, New York: McGraw-Hill, 1984.

Vogt88 Vogt, C.F. "Group Technology: CIM's Sleeping Giant." *Design News* 22-23, 1988.

Wang81 Wang, K.L. & Cheng, H.S. "Numerical Solution to the Dynamic Load, Film Thickness, and Surface Temperatures in Spur Gears–1, 2." *ASME Journal of Mechanical Design* 103(4), Jan. 1981, p.177-194.

Wein87d Weinrib, J.M. "John Kenneth Galbraith: 'Beyond Technology Comes Good Design...'" *Mechanical Engineering* 109(1):50-52, 1987.

Wins92 Winston, P. *Artificial Intelligence*, Reading, MA: Addison-Wesley, 1992. Ed. 3

Yode93 Yoder, M. "Diamond Properties and Applications." In: *Diamond Films and Coatings: Development, Properties, and Applications*, edited by Davis, R. Park Ridge, NJ: Noyes Publications, 1993, p. 1-30.

Zimm86 Zimmerman, J. "Clutches and Brakes." In: *Standard Handbook of Machine Design*, edited by Shigley, J. & Mischke, C., New York: McGraw-Hill, 1986, p. 30.1-30.51.

BIOGRAPHICAL SKETCH

The author received his Bachelor of Science in Mechanical Engineering at the University of Florida in December 1985. He worked for five and a half years for United Technologies—Pratt & Whitney in West Palm Beach, Florida, as a design engineer. During the author's career at Pratt & Whitney, he completed his Master of Engineering degree in mechanical engineering through a distance learning program. He presented his master's project, "Fan Drive Turbine Disk Synthesis and Optimization," and graduated in the Spring of 1991. In August of 1991, he took advantage of an early retirement separation package at Pratt & Whitney to pursue his doctorate.

In the time between leaving Pratt & Whitney and beginning his doctoral studies, he met the love of his life. In the Fall of 1992, he moved his bride Linda and their adopted dog, Mr. Murphy, to Gainesville. The author and his family plan to continue leading a happy and fulfilling life.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Ali A. Seireg, Chairman
Ebaugh Professor of Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Carl D. Crane, III
Associate Professor of Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Douglas D. Dankel, II
Assistant Professor of Computer and Information Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

D. Jack Elzinga
Professor of Industrial and Systems Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Rajiv K. Singh
Associate Professor of Materials
Science and Engineering

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as a partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 1995

Winfred M. Phillips
Dean, College of Engineering

Karen A. Holbrook
Dean, Graduate School